オーバーレイネットワークにおける分散認証に関する研究

平成 21 年度　修士学位論文

オーバーレイネットワークにおける

分散認証に関する研究

東北大学大学院情報科学研究科 情報基礎科学専攻

博士課程前期 2 年の課程

コミュニケーション論講座 (白鳥研究室)

A8IM1025　中山 誠也

目 次

図 目 次

表 目 次

第1章　序論

本章では，ユビキタス情報環境における個人情報の活用について説明し，ユビキタス情報環境において，個人情報の保護を行う際に起こりうる問題点について述べる．続いて，本研究の目的と本研究による効果について挙げ，最後に本論文の構成について述べる．

1.1　研究の背景

近年，小型携帯端末の高度化や低価格化，センシング技術の発達により，ユビキタス情報環境の実現可能性が高まっており，ユーザの日常生活を支えることが期待されている [1]．ユビキタス情報環境におけるアプリケーションとして，利用者の個人情報などを用いることで，個人のスケジュール管理やヘルスケアを行うサービスの研究開発が盛んに行われている [2]．このような環境においては，利用者の健康情報や個人情報などのプライバシーに関わる情報が，端末間で頻繁にやりとりされることが必須となるため，情報の盗聴や改ざん，端末のなりすましなどといった行為を技術的に防止することが重要となる．

情報を保護するために広く用いられている技術として，公開鍵認証基盤 (PKI：Public Key Infrastructure)[3] がある．PKI では認証局（CA：Certificate Authority）と呼ばれるサーバが各利用者端末（以下，ノードと呼ぶ）の公開鍵に対して公開鍵証明書を発行する．各ノードは必要に応じて，通信相手の公開鍵証明書に付加された電子署名を CA の公開鍵で検証することで，通信相手の公開鍵の正当性を確認する．PKI では，ノードの利用者と CA の管理者との社会的な信頼関係に基づいて公開鍵証明書の発行を行うため，公開鍵証明書を取得するためには複雑な手続きと金銭的な対価が必要とされる．世界中で最も広く利用され，事実上のスタンダートなっている CA を運営している証明書発行組織として VeriSign, Inc.[5] が存在するが，この日本法人である日本ベリサイン株式会社 [6] から 1 年

間有効のサーバ証明書を取得するためには，85,050 円の代金を支払うことが必要となる（2009 年 12 月現在）．そのため，個人が所有するために，気軽に証明書を取得することは出来ない．よって，多数のクライアント端末の認証を行う必要があるユビキタス環境においては，より低コストに利用可能な認証手法が必要となる．

また，認証局はすでに無効となっている証明書を失効証明書リスト（CRL：Certificate Revocation List）[4] として一元管理しており，各ノードは認証局の署名が付与された証明書を受け取るたびに，その証明書が CRL に含まれていないかを CA に問い合わせる必要が生じてしまい，CA に負荷が集中してしまうため，スケーラビリティにも問題がある．そのため，膨大な数のノードがネットワークに参加することが予想されるユビキタス情報環境へ PKI を適用することは困難である．

この様な背景から，ユビキタス情報環境では誰もが気軽に安全なネットワークを利用するための新たな仕組みが必要である．そこで，本研究では，ユビキタス情報環境において固定的な機関やサーバを用いることなく，複雑な手続きを必要としない認証基盤を実現することを目的とする．

1.2　本研究の目的

図 1.1 にユビキタス環境において実現されるサービスの例を示す．近い将来実現されることが予測されるユビキタスサービスにおいては，利用者の個人情報が積極的に利用されることが予測される．例えば，バス停で足が不自由な高齢者がバスを待っている場合，バスが停留所に到着する前にあらかじめ乗客に席を譲っておいてもらうことをアナウンスしておくことで，相手に余計な気を使うことなく気配りを行うことが可能となる．また，カロリーを気にしている人に，高カロリーのメニューを避けて低カロリーなメニューを推薦するなどといったことも可能となる．このようなサービスは，個人の持つ健康情報や嗜好の情報を用いることで実現される．

こういったサービスを提供する際に利用される，健康情報や個人の趣味嗜好，所属に関わる情報などは，不用意に流出しまうと悪用される危険が高い．そのため，これらの情

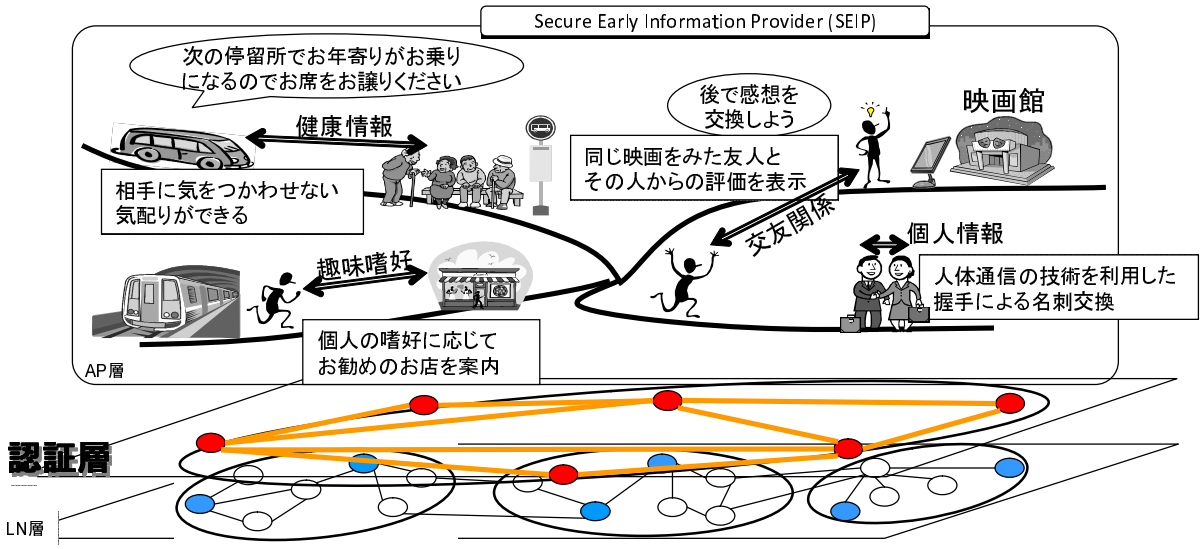

図 1.1: ユビキタス環境におけるサービス

報のやりとりを行う際には，セキュリティに配慮し，データの盗聴や改ざんを防止することが重要となる．しかし，ユビキタス情報環境においては，非常に多数のノードがネットワークに参加することが予測されるため，固定的なサーバに負荷が集中する手法や，利用する際に複雑な手続きが必要となってしまう手法は適切ではない．

そこで本研究では，ネットワークに参加するノード全体で分散認証を行うことを目的としたオーバーレイネットワーク構成法の確立を目的とする．このオーバーレイネットワークを利用する事により，複雑な手続きを必要とすることなく，安全な通信が提供可能なスケーラブルな認証基盤が実現できる．具体的には，ネットワークに参加する各ノードがお互いの公開鍵を分散管理し，必要に応じて各ノードが任意のノードの公開鍵を確実に取得できるようにする．ここで通信の安全性を保障するために，各ノードはノード間の信頼関係に基づき公開鍵を取得する必要がある．また，低コストでかつ容易に使用することを可能とするためには，各ノードが公開鍵の登録や削除，検索を自動的に行えなければならない．さらに，ユビキタス情報環境においては，ネットワークが非常に大規模に展開されることが予測されるため，それらに対応できるスケーラビリティを確保することが重要な要件となるが，その際，携帯端末などの PDA やセンサデバイスなどの比較的低性能な端末

でも参加が可能である必要がある．

これらの要求に対し，本論文では，ネットワークへの参加ノードをその利用可能な CPU 計算能力などの計算機資源に応じて 2 段階に階層化を行うことで，低性能端末でも参加可能となる分散認証手法を提案する．そして，計算機シミュレーションを通して，本手法が，各端末の計算機資源の状況を動的に判別することで，認証のための計算機資源に余裕がないノードの参加にも対応可能な認証ネットワークを形成可能であることを示す．

1.3 本研究の効果

本節では，提案手法を用いて分散認証基盤を実現した際に得られる効果について説明する．ユビキタス情報環境において分散認証を用いるサービスの例として，図 1.2 に示すような「携帯端末からの戸締りサービス」を取り上げる．このサービスは，認証ネットワークを適用することにより得られるサービスの一例であり，外出先から自宅の扉の鍵を確認・施錠出来るというものである．自宅の扉の開閉情報が開示されるべき本人以外に漏洩してしまうと，その情報が悪用される恐れがあるため，公開鍵暗号を用いて情報を安全に送受信することが求められる．

まず，図 1.2(a) に，携帯端末からの戸締りサービスに PKI を適用した例を示す．PKI では，信頼できる第 3 者機関である認証局が全ての端末に公開鍵証明書を発行するため，極めて多数の端末がネットワークに参加するユビキタス情報環境では，認証局と各端末との通信にかかるコストが膨大となってしまう．すなわち，PKI は，スケーラビリティの低い手法である．

次に，図 1.2(b) に，携帯端末の戸締りサービスに既存の分散認証を適用した例を示す．既存の分散認証手法においては，ネットワークに参加する端末が連携して公開鍵の配布を行うため，負荷分散が可能となり，スケーラビリティの課題を解決できる．しかし，すべての端末に平等に一定の負荷が生じてしまうため，携帯端末などの低性能端末に過剰な負荷が掛かる．そのため，適切な公開鍵の配布が行われなくなり，認証ネットワーク全体が破綻してしまうという問題がある．すなわち，携帯電話やセンサデバイスなどといった多

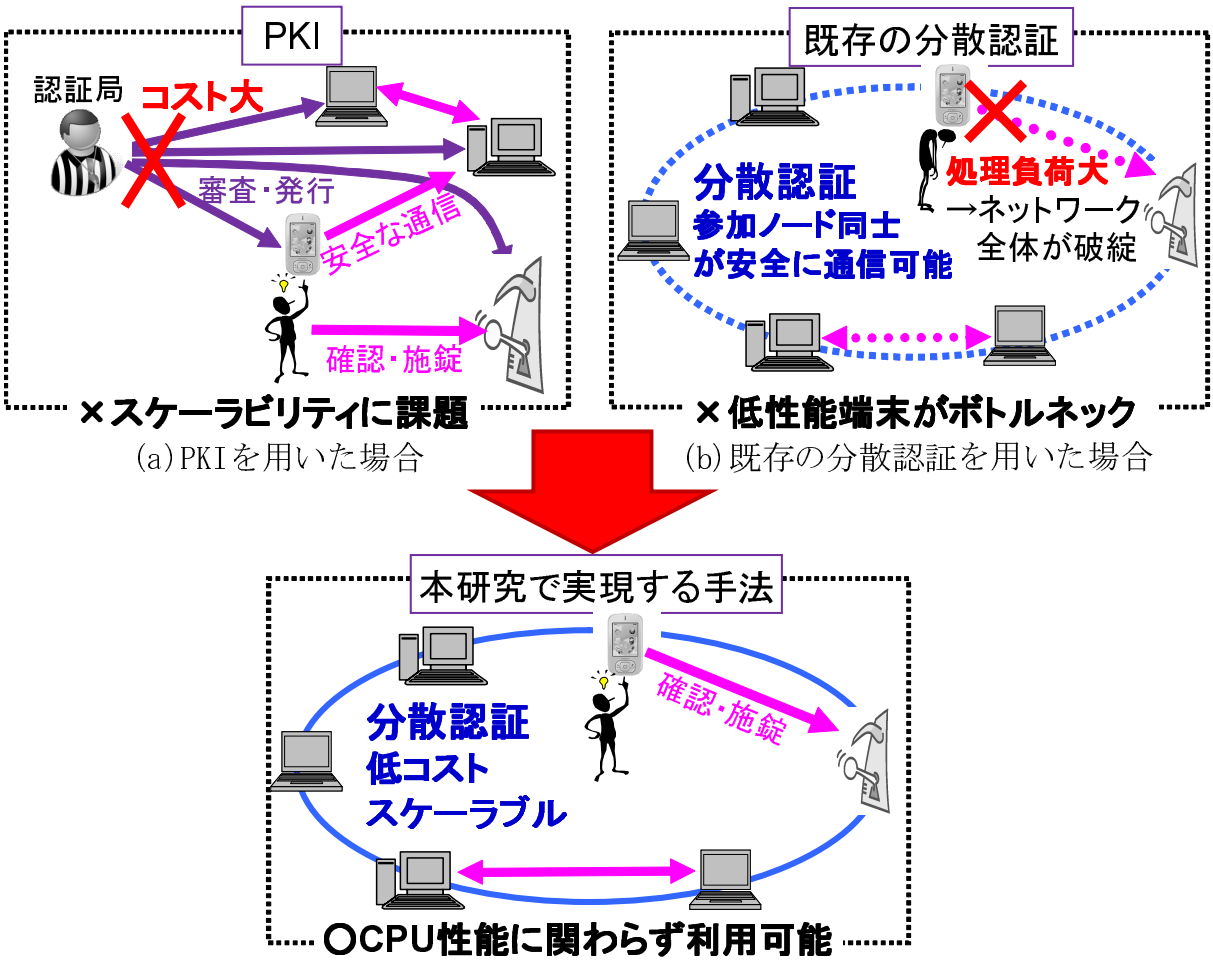

図 1.2: 本研究の効果

様な携帯端末がネットワークに参加するユビキタス情報環境に，既存の分散認証手法を適用することは難しい．

図 1.2(c) に，携帯端末からの戸締りサービスに提案手法を適用した例を示す．提案手法においては，スケーラブルで低コストな分散認証を実現すると同時に，携帯端末などの処理能力の低い端末に生じる負荷を著しく軽減することにより，CPU などの資源状況に関わらず利用可能な分散認証基盤の実現が可能となる．

1.4　本研究の位置づけ

コンピュータ端末の認証手法における本研究の位置づけを図 1.3 に示す．認証手法における目標は，セキュリティの向上と認証にかかるコストの削減に大きく分けることが出来る．例えば，PKI は審査に基づいた高い信頼性を保障出来るが，公開鍵証明書の発行にかかる時間や費用などのコストが大きい．よって，クライアントの認証は行わず，少数のサーバ認証のみを行えればよい電子商取引などの分野での利用に適している．しかし，この手法は，多数のクライアント端末の認証が必要となってくる分野において使用することは適切でない．

クライアント認証を行う手法として，OpenID[7] やペルソナカード [8] がある．これらの技術では，クライアントの同一性の確認を行うことで，ユーザのなりすましを防ぐことが可能となる．しかし，通信データに暗号化やデジタル署名を施したりすることは不可能であるため，通信を行う端末間で行われる中間者攻撃などを防ぐことが出来ない，そのため，これらの技術だけでは，データの盗聴や改ざんが行われてしまう危険性がある．

本研究で実現を目指す手法は，ユーザのなりすましや，通信データの盗聴や改ざんを防ぐことを目的としながらも，PKI における CA のような第 3 者機関を必要としないものである．すなわち，図 1.3 の Proposal と示された位置づけとなる．

1.5　本論文の構成

本論文は，以下のような構成となっている．第 2 章において関連研究を紹介し，オーバーレイネットワークにおける分散認証における問題点を述べる．次に，第 3 章において，ネットワークに参加するノードの階層化を動的に行い，低性能端末への負担を最小限に抑える分散認証手法（HiHDAM：Hierarchical Hash-based Distributed Authentication Method）を提案する．その後，第 4 章で計算機シミュレーションを通して提案手法の有効性を評価する．最後に，第 5 章でまとめと今後の課題について述べる．

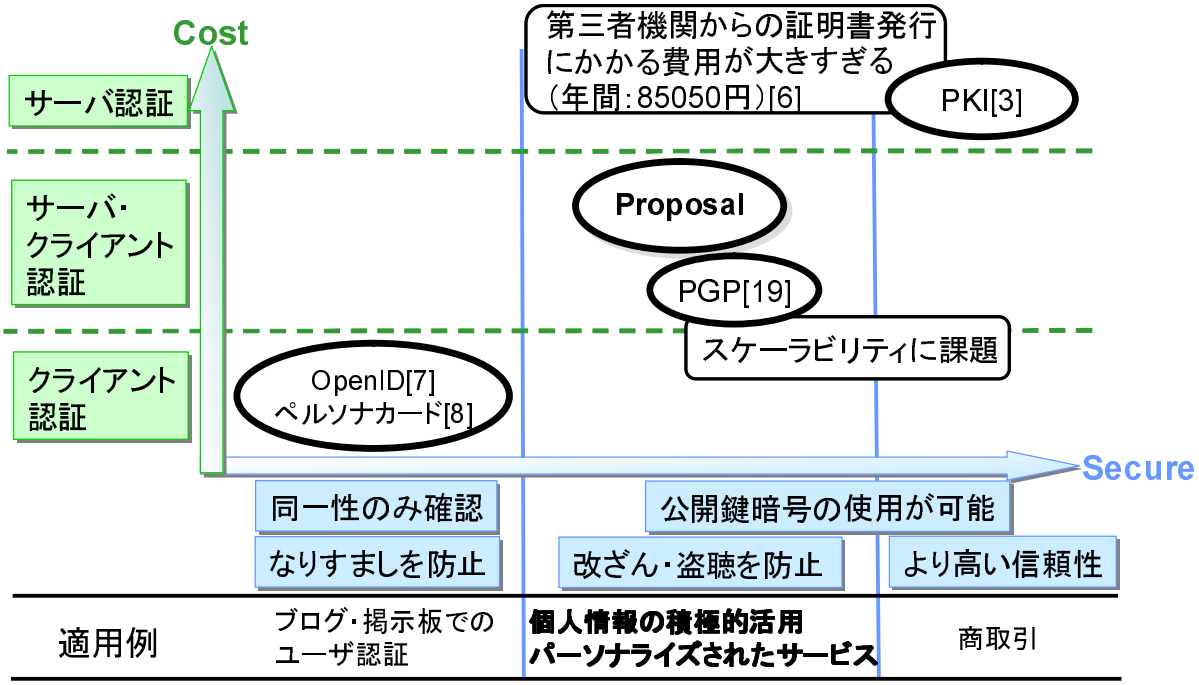

図 1.3: 認証手法における本研究の位置づけ

第2章　オーバーレイネットワークにおける関連研究と課題

本章ではまず，オーバーレイネットワークの構成方法の分類について説明した後，オーバーレイネットワークにおけるセキュリティに関する技術について，その分類を説明し，その分類における本研究の位置づけを述べる．最後に，本研究と近い位置づけにある既存研究を詳しく紹介し，オーバーレイネットワークのセキュリティにおける問題点について述べる．

2.1　オーバーレイネットワークの構築方法

オーバーレイネットワーク技術はアプリケーションレベルのネットワークを論理的に構成することで，ノード間の直接通信によるルーティングや検索などを可能とする技術である．オーバーレイネットワークは，物理層ネットワークとは独立に構成され，その構造から構造型オーバレイネットワークと非構造型オーバーレイネットワークに大別出来る．本節では，これらの特徴について説明し，ユビキタス情報環境において必要とされる分散認証において，達成すべき目標を述べる．

2.1.1　非構造型オーバーレイネットワーク

非構造型オーバーレイネットワークにおいては，ネットワークに参加するノードは論理的に無秩序に配置され，データの検索にフラッディング探索が用いられる．フラッディング探索では，ノードが目的とするデータを検索する際，論理的に直接接続されているノー

ドにクエリをブロードキャストする．クエリを受け取ったノードは自身が対象とされているデータを保持している場合は，送信元のノードと end-to-end で通信を行いデータの送受信を行うが，対象となるデータを保持していない場合は，送信元のノードと同様に論理的に接続されているノードにクエリをブロードキャストする．制限を設けず，フラッディング探索を行った場合，クエリが無限にネットワークで送信され続ける可能性が生じるため，クエリには通常，送信元ノードが Time to Live（TTL）を付加する．非構造型オーバーレイネットワークにおいては，TTL の値が大きくなるほど目的のデータの検索に成功する可能性は向上するが，ネットワークに生じるクエリの数も増大してしまう．また，TTL の値が小さくなれば，ネットワークに生じるクエリ数は減少するが，データの検索の成功率が低下してしまう問題が生じる．非構造型オーバーレイネットワークを用いたアプリケーションの主な例として，Napster[9]，Gnutella[10]，BitTorrent[11]，Winny[12] などがある．

2.1.2 構造型オーバーレイネットワーク

構造型オーバーレイネットワークにおいては，ネットワークに参加するノードを論理的に配置する場所や，データの検索方法に明確なプロトコルが定められている．具体的には，各ノードが特定のノードの論理位置情報などが記録された分散ハッシュテーブル（DHT：Distributed Hash Table）を管理し，その情報をもとにプロトコルに従いクエリを伝送することで，クエリがネットワーク中に増大してしまうことを防ぐのと同時に，高い検索成功率を実現している．構造型オーバーレイネットワーク技術の代表例として，Chord[13]，CAN(Content Adressable Network)[14]，Skip Graph[15]，Pastry[16]，Tapestry[17] などがある．

従来のこれらの技術においては，画像や音声，映像などといったコンテンツの共有が目的とされており，オーバーレイネットワーク上において安全な通信を実現することに関しては重要とされていなかった．本研究においては，オーバーレイネットワーク上において行われる通信が安全なものであることを保証することを最優先とし，その上で発生するク

エリ数や各端末に生じる負荷を削減することを目標とする．

2.2 オーバーレイネットワークにおけるセキュリティ

オーバーレイネットワークにおけるセキュリティに関する研究は，以下の 3 種類に大別することが出来る [18].

1. Web of Trust
2. Statistical Trust
3. Hybrid Trust

このうち，Statistical Trust[27],[28] と Hybrid Trust[29],[30] においては，ネットワークに参加するノードの投票結果などに基づき，信頼可能なノードを判断することで，ネットワーク内に悪意あるノードが参加することを防止することを目的としている．一方，Web of Trust においては，ユーザー同士が互いに他者の公開鍵に対して電子署名を行うことで，公開鍵とその所有者との結び付きを保証する．そのようにして構築されたノード間の信頼関係に基づき，各ノードが必要となる公開鍵を取得することで，通信データに暗号化やデジタル署名を施すことが可能となる．

本研究においては，ユビキタス情報環境における個人情報等を積極的に活用したサービスを支援することを目的とするため，データの暗号化やデジタル署名を行うのに不可欠である Web of Trust を対象とする．

2.3 Web of Trust

Web of Trust では，ユーザー同士が互いに他者の公開鍵に対して電子署名を行うことで，公開鍵とその所有者との結び付きを保証する．そのようにして信頼性が保証された公開鍵を各ノードが用いることで，ノード間の信頼関係に基づき必要となる公開鍵の取得を行うことができ，通信データに暗号化やデジタル署名を施すことが可能となる．

図 2.1: ノードの認証

Web of Trust において，ネットワークに参加するすべてのノードは，自身で秘密鍵と公開鍵のペアを作成する．このとき，公開鍵は広く公開し，誰にでも入手可能な状態であるものとする．一方，秘密鍵は自身以外のノードには譲渡や閲覧を決して許可しないものであるとし，確実に自身のみが閲覧・使用できるものとする．ここで，2 つのノード A,B が存在したとする．B は自身の秘密鍵を用いることで，通信データに電子署名を付加することが可能となる．このとき，A が B の公開鍵 K_B を保持している場合，A は B からの電子署名を検証することで A の本人性を確認することが可能となる．よって，本研究では A が B の公開鍵 K_B を保持している状態を ”A が B を認証している状態 ”と呼ぶ．また，A が認証しているノードの集合を $A.trust$ とあらわす．図 2.1 は，ノード A がノード B を認証可能である状態を示したものである．

図 2.2 に信頼の輪に基づくノードの認証手順を示す．ノード A，B，C，D が存在し，$B \in A.trust$，$C \in B.trust$，$D \in C.trust$ であり，D が A に対して認証要求を行った場合を考える．このとき，A は D の公開鍵 K_D を保持していないため，D を認証することが出来ない．そこで，以下の手順に従って公開鍵 K_D を入手する．

1. B から K_C を入手する．$C \in A.trust$ となる．

2. C から K_D を入手する．$D \in A.trust$ となる．

以上の手順により，A は D の公開鍵 K_D を入手し，K_D を用いて D を認証することが可能となる．このように，認証済みのノードから間接的に公開鍵を入手して新たなノードの認証を行う認証手法を信頼の輪を用いた認証と呼ぶ．

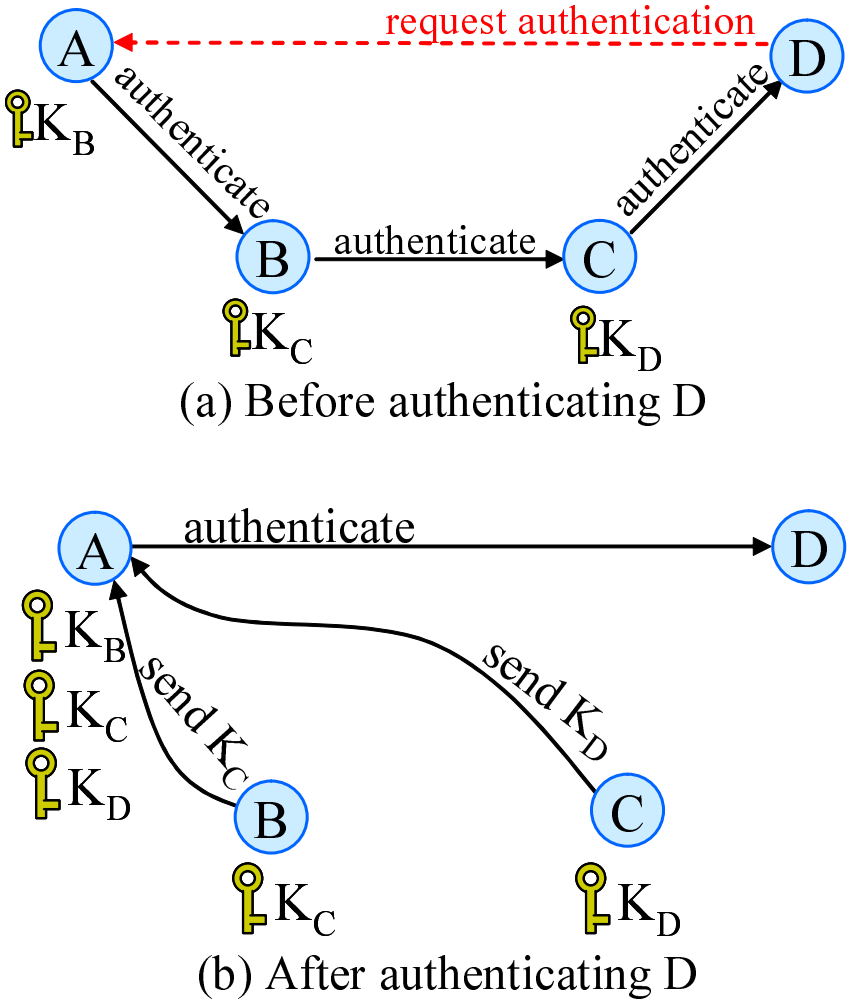

図 2.2: 信頼の輪を用いたノードの認証

2.4 Web of Trustに関する既存研究

Web of Trust をはじめて実現したのは，Pretty Good Privacy(PGP)[19] という暗号ソフトウェアである．PGP では，信頼の輪と呼ばれる各ノードの利用者間の信頼関係を活用することにより，信頼できるノードを介して新たな公開鍵を収集することで，認証局を必要としない分散型の公開鍵管理を実現している．そのため，ノードの利用者と認証局の管理者との社会的な信頼関係の確認も不要となり，各ノードの利用者間の合意によって新たな公開鍵の登録をすることが可能となる．しかし，PGP は新たな公開鍵を検索するための仕組みを持たず，計画的な信頼の輪を形成することが出来ない．ユビキタス情報環境においては，多数の端末がネットワークに参加するため，必要に応じて目的の公開鍵だけを取得することが望ましいため，PGP を適用することは適切でない．

また，各ノードが同じネットワークに参加する全てのノードの公開鍵を自律的に収集する手法が提案されている [20]．この手法では，各ノードが自律的に全ノードの公開鍵を収集することにより，サーバを利用せずにノード間の認証を実現している．しかし，この手法は全ての公開鍵を無計画に収集するため，各ノードは多くのメモリ量と通信データ量を必要とする．

サーバを必要とせず，かつ効率的に公開鍵の管理と配布を実現する公開鍵分散管理手法が提案されている [21],[22]．これらの手法では，ネットワークのルーティング情報などに基づき新しい公開鍵を自動的に収集することで，ノードが必要とするメモリ量と通信メッセージ量を削減している．しかし，ルーティング情報などのネットワーク特有の情報を利用するため，適用できるネットワークが限定されてしまうという問題がある．

適用可能なネットワークを限定せずに，効率的に公開鍵の分散管理を実現することを目的とした手法がある [23],[24],[25],[26].

このうち，ノードが自律的に Trust Group (TG) を形成して，自身の TG に所属するノードに対して必要な公開鍵を要求する手法 [23] においては，TG の個数の設定によって，問題が生じてしまう．TG の個数が少なすぎる場合は，必要な公開鍵が手に入らない可能性が増大してしまう．逆に TG の個数が大きすぎる場合では，TG に所属するすべてのノー

ドに検索を要求するメッセージが送信されるため，各ノードに生じる負担が増加してしまう．また，TG となるノードは無作為に決定されるため，TG となるノードの個数を増加させたとしても，求める公開鍵の取得に失敗する可能性が常にあるという欠点がある．

Super-peer が社会的な信頼関係を用いてフルコネクト型で結合され，Super-peer が registerd-peer を 1 対多の関係で管理する手法 [24],[25] では，すべての Super-peer 間が論理的にフルコネクト型で結合されるので，Super-peer への負担が大きいという課題や，新しく Super-Peer が参加する際に Super-peer 間でやりとりされるメッセージ数が膨大なものとなってしまうという問題がある．

Hash-based Distributed Authentication Method(HDAM)[26] では，信頼の輪と分散ハッシュテーブル（DHT)[13] を用いることで，確実な公開鍵の管理・配布を実現している．しかし，参加するすべてのノードは，ほかのノードによる公開鍵の取得の中継を行わなければならず，すべてのノードに一定の負荷が生じてしまうため，センサデバイスや携帯端末などの低性能な計算機端末の参加に対応することが難しくなっている．そのため，多様な計算機端末の参加が予想されるユビキタス情報環境へ HDAM を適用することは難しい．また，HDAM においてはネットワークに参加する複数のノードによって公開鍵の中継が行われるが，そのすべてのノードがプロトコルに従って正しく動作するという前提に基づいて設計がされており，悪意を持って公開鍵の中継の妨害を行うノードの参加を考慮していないという課題もある．

よって本研究では，ネットワークに参加するノードの階層化を行うことにより，公開鍵暗号を復号する処理を連続して行うには計算能力が不充分な低性能端末や，アプリケーション上での処理によって計算機資源 に高い負荷が生じているノードでも参加可能な公開鍵分散認証方式を実現する．また，提案手法においてはネットワークに悪意あるノードが存在する場合においても，高い確率で必要な公開鍵を入手できるよう工夫を施す．表 2.1 に既存研究と本研究で実現する手法についてまとめたものを示す.

表 2.1: Web of Trust における関連研究

	信頼性	ネットワーク制約	検索要求到達	管理効率	低性能端末の参加
PGP[19]	高◎	無し○	不確実×	低い×	困難×
S-O-M[20]	中○	無し○	確実◎	低い×	困難×
ASNS[21] DecentCA[22]	中○	有り×	確実○	高い◎	可能△ （ただし現実的でない）
BFTA[23]	中○	無し○	不確実×	高い○	困難×
Trusted web[24] P2P-PKI[25]	中○	無し○	確実◎	低い×	可能◎
HDAM[26]	中○	無し○	確実△ （アタッカー対策無）	高い○	困難◎
提案手法	中○	無し○	確実○	高い○	可能◎

第3章　提案：階層型公開鍵分散認証方式

本章では，本研究における目的を達成するため，2 章で提起した技術的課題を解決するための「階層型公開鍵分散認証方式（HiHDAM：Hierarchical Hash-based Distributed Authentication Method）」について述べる．

3.1　提案方式の概要

ユビキタス情報環境において公開鍵暗号を用いた安全な通信を行うためには，利便性に優れ，スケーラブルであることに加え，センサデバイスや携帯端末などの低性能端末でも参加できる公開鍵認証方式が必要である．そこで本研究では，ネットワークに参加している任意のノード間の認証と公開鍵の効率的な分散管理を実現する HDAM に親ノード・子ノードの概念を取り入れることで，低性能端末でも参加可能であるスケーラブルな公開鍵分散認証方式を提案する．

提案手法では，効率的に公開鍵を分散管理するために仮想的にオーバーレイネットワークを構築する．その際，高性能端末を親ノードとし，ハッシュリング上に配置する．親ノードは DHT を効果的に用いて信頼の輪を形成することにより，公開鍵の安全で効率的な分散管理を実現する．また，低性能端末は子ノードとし，ハッシュリング上に配置せず，親ノードを介して必要な公開鍵を入手できるようにする．これにより，子ノードはハッシュリング上で行われる公開鍵の分散管理や公開鍵の配布の中継を行わなくてよいため，それに伴う暗号の復号化や電子署名の検証を行う必要はなくなり，子ノードの負担を最小限に抑えることが可能となる．

また，構築したオーバーレイネットワーク中に悪意のあるノードが存在している場合においても，複数の経路を用いた検索を行うことで公開鍵の検索の成功率を向上させる．

3.2 定義

本研究における変数の定義を，表 3.1 に示す．本研究におけるノードの認証とは，通信データに付加された電子署名と対象ノードの公開鍵を用いてノードの本人性を認証することを指す.

3.3 公開鍵の分散管理

図 3.1 に HiHDAM による公開鍵の分散管理の例を示す．ここで，A〜G は親ノードを示し，a1,e1,e2 はそれぞれノード A とノード E の子ノードを示す．また，$X.hash$ は親ノード X のハッシュ値，$x.parent$ は子ノード x の親ノード，K_i はノード i の公開鍵をそれぞれ示す．親ノードはそれぞれの識別子から一方向ハッシュ関数で決められたハッシュ値を基に，1 から N までの指標を円形に配置したハッシュリング上に仮想的に配置される．子ノードは，同様の方法で得られたハッシュ値を基に決められた親ノードを介して HiHDAM に参加する．親ノードはハッシュリングにおいて自身の位置から正の方向に $2^k(k = 0, 1, 2, \ldots$ 以上離れたノードのうち最も近い位置に配置されたノードの公開鍵と，自身の子ノードの公開鍵を管理する．ネットワークに参加するノード数が n のとき，各親ノードが管理する

表 3.1: 変数定義

変数	定義
K_i	ノード i の公開鍵
$X.hash$	親ノード X のハッシュ値
$x.parent$	子ノード x の親ノード
N	ハッシュ値がとりうる最大の値（最大ハッシュ値）

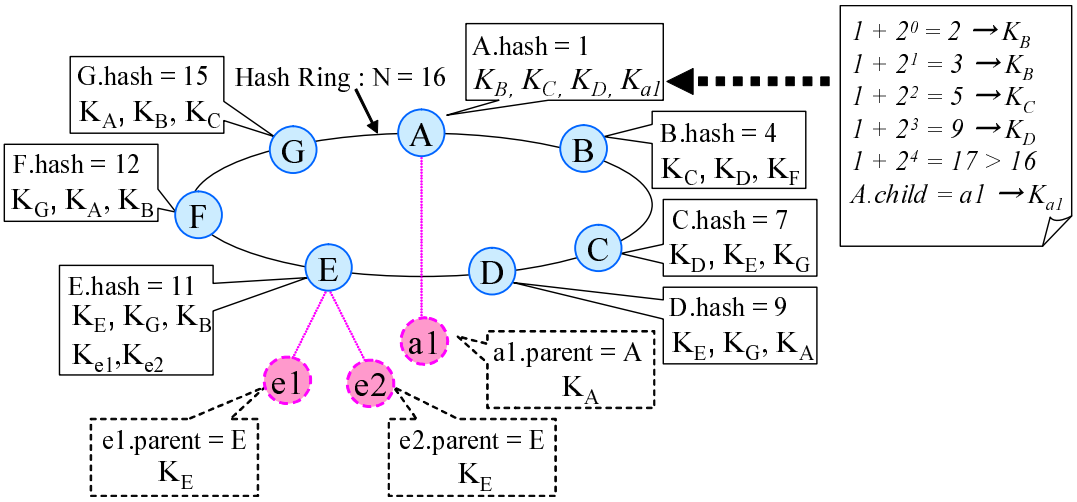

図 3.1: 公開鍵の分散管理

公開鍵の最大数は $\log n$ となる．図 3.1 の場合，A が管理する公開鍵は以下の 4 個となる．

- 正の方向に $2^1(2^0)$ 以上離れたノードの中で最も近い位置に配置されたノードである B の公開鍵 K_B
- 正の方向に 2^2 以上離れたノードの中で最も近い位置に配置されたノードである C の公開鍵 K_C
- 正の方向に 2^2 以上離れたノードの中で最も近い位置に配置されたノードである D の公開鍵 K_D
- 自身の子ノードである a1 の公開鍵であるノード a1 の公開鍵 K_{a1}

また，子ノードは自身の親ノードの公開鍵のみを管理することとなり，図 3.1 の場合，a1 が管理する公開鍵は自身の親ノードである A の公開鍵 K_A1 個のみとなる．

3.4 信頼の輪と DHT を用いた公開鍵の取得

子ノード s が子ノード d の公開鍵を保持していない状態で，d から s への認証要求が行われるなどし，公開鍵の入手が必要になった場合，以下の手順により，s は d の公開鍵を取得する．

1. s は自身の親ノード S に d の公開鍵 K_d を要求する．
2. S が公開鍵 K_d を保持している場合，S は s へ K_d を送信する．
3. S が公開鍵 K_d を保持していない場合，S は d の親ノードである D の公開鍵 K_D を保持しているか確認する．
   - 3.1. S が公開鍵 K_D を保持している場合，S は D に対し，d の公開鍵 K_d を要求する．
   - 3.2. S が公開鍵 K_D を保持していない場合，S は自身が公開鍵を保持しているノードの中から，ハッシュリング上で最も D に近い位置に配置され，かつ D より小さいハッシュ値を持つノード S^t に対し，D の公開鍵 K_D を要求する．
   - 3.3. S^t が公開鍵 K_D を保持している場合，S^t は S へ公開鍵 K_D を送信する．
   - 3.4. S^t が公開鍵 K_D を保持していない場合，S^t が公開鍵を保持しているノードの中から，ハッシュリング上で最も D に近い位置に配置され，かつ D より小さいハッシュ値を持つノード S' の公開鍵 $K_{S'}$ を S に送信する．S は S' の公開鍵 $K_{S'}$ を入手し，手順 3.2 へ戻る．
4. S は D に対し d の公開鍵 K_d を要求する．
5. D は S に対し，d の公開鍵である K_d を送信する．
6. S は受信した公開鍵 K_d を s へ送信する．

また，親ノード S が子ノード d の公開鍵を入手する場合には，上で示した手順 3 から手順 6 までの動作を行う．親ノード S が親ノード D の公開鍵を入手する場合には，手順 3.2 から手順 3.4 までの動作を行うことで解決できる．同様に，子ノード s が親ノード D の公

開鍵を入手する場合には，手順1から手順3までの動作を通してsの親ノードであるノードSがノードDの公開鍵K_Dを入手した段階で，sに対し公開鍵K_Dを送信することで解決できる．ネットワークに参加するノード数がnのとき，公開鍵の入手に必要となる通信データ量は$O(\log n)$となる．その際，ネットワークで送受信されるメッセージの大部分を占める公開鍵の中継に関してはすべて親ノードが行い，子ノードがメッセージを処理しなければならないのは，自身の公開鍵を要求された場合と，自身が要求した公開鍵が送信されてくる場合のみとなる．これにより子ノードの負担を最小限に抑えることができ，低性能端末でも参加可能な分散管理を実現できる．

図3.2にノード間の認証手順の例を示す．この例では，前述した手順に従い，図3.1におけるノードe1がノードa1を認証する．具体的には以下の手順となる．

1. e1は自身の親ノードEにa1の公開鍵K_{a1}を要求する．
2. Eが公開鍵K_{a1}を保持している場合，Eはe1へK_{a1}を送信する．
3. Eが公開鍵K_{a1}を保持していない場合，Eはa1の親ノードであるAの公開鍵K_Aを保持しているか確認する．
   - 3.1. Eが公開鍵K_Aを保持している場合，EはAに対し，a1の公開鍵K_{a1}を要求する．
   - 3.2. Eが公開鍵K_Aを保持していない場合，Eは自身が公開鍵を保持しているノードの中から，ハッシュリング上で最もAに近い位置に配置され，かつAより小さいハッシュ値を持つノードS^tに対し，Aの公開鍵K_Aを要求する．
   - 3.3. S^tが公開鍵K_Aを保持している場合，S^tはEへ公開鍵K_Aを送信する．
   - 3.4. S^tが公開鍵K_Aを保持していない場合，S^tが公開鍵を保持しているノードの中から，ハッシュリング上で最もAに近い位置に配置され，かつAより小さいハッシュ値を持つノードS'の公開鍵$K_{S'}$をEに送信する．EはS'の公開鍵$K_{S'}$を入手し，手順3.2へ戻る．
4. EはAに対しa1の公開鍵K_{a1}を要求する．

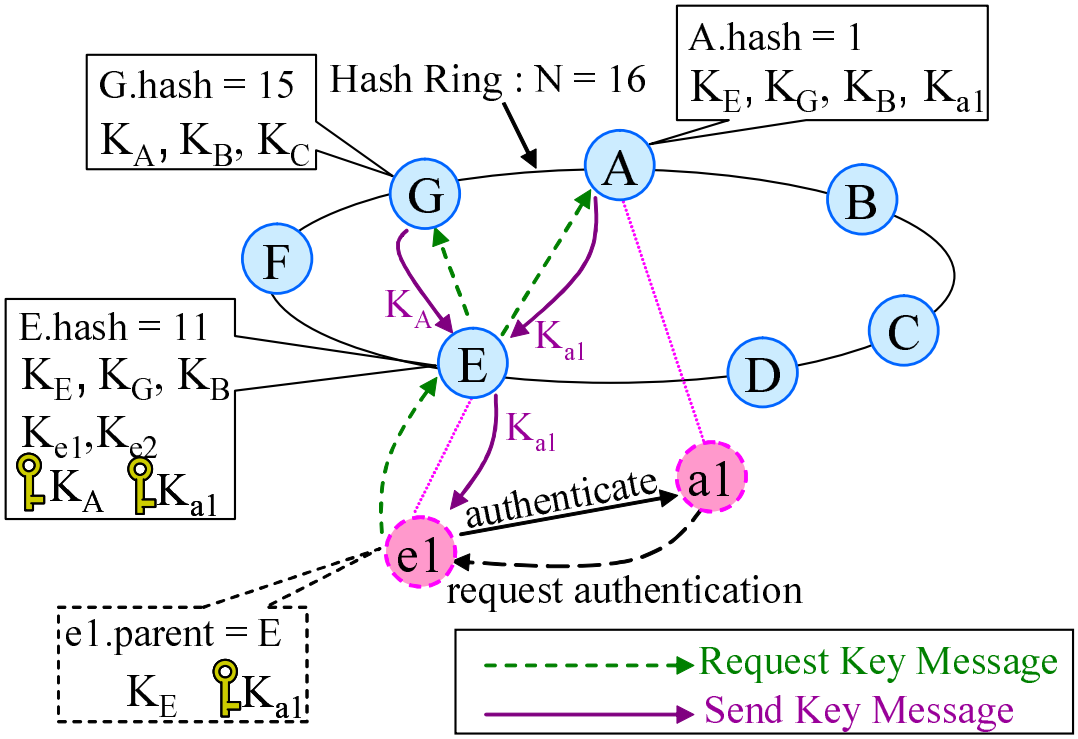

図 3.2: 公開鍵の取得

5. A は E に対し，a1 の公開鍵である K_{a1} を送信する．

6. E は受信した公開鍵 K_{a1} を e1 へ送信する．

以上の手順により，e1 は公開鍵 K_{a1} を入手し，a1 を認証する．

3.5 ノードの参加手順

ノード n が HiHDAM に参加するときの前提条件として，n はすでに HiHDAM に参加している任意のノード g と公開鍵を交換済みであるものとする．n が親ノードだった場合の HiHDAM に参加する手順を以下に述べる．

1. n はハッシュリング上で自身の正側に隣接するノード $n.successor$ の公開鍵を g から取得する．

2. n が認証すべきノードを計算し，それらの公開鍵を $n.successor$ から取得する．

3. $n.successor$ は自身の公開鍵を配布したノードに対して，信頼の輪の再構築を通知する．

4. 再構築の通知を受けたノードは自身が認証すべきノードを再計算し，それらのノードの公開鍵を取得する．n の公開鍵は $n.successor$ から取得する．

n が子ノードだった場合は，n は g を通して n の親ノード $n.parent$ の公開鍵を取得することで HiHDAM に参加する．その際，n の親ノード $n.parent$ は，一方向性ハッシュ関数を用い n の持つ固有の情報から一意に決定される．

図 3.3 に親ノードが HiHDAM に参加する際の例を示す．この例では，親ノード Z がノード B を介してネットワークに参加する．以下，本例をもとに処理手順を述べる．

1. Z はハッシュリング上で自身の正側に隣接するノード G の公開鍵 K_G を B から取得する．

2. Z は自身が認証するノード G,A,C の公開鍵を G から取得する．

3. G は自身の公開鍵を保有しているノード F,E,D,C に対して，信頼の輪の再構築を通知する．この通知をうけたノードのうち，F,E,D は Z を認証するために Z の公開鍵 K_Z を G から取得する．

以上の手順により親ノード Z は HiHDAM へ参加する. ネットワークに参加するノード数が n のとき，新しい親ノードがネットワークに参加するために必要な通信データ量は $O(\log n)$ となる．

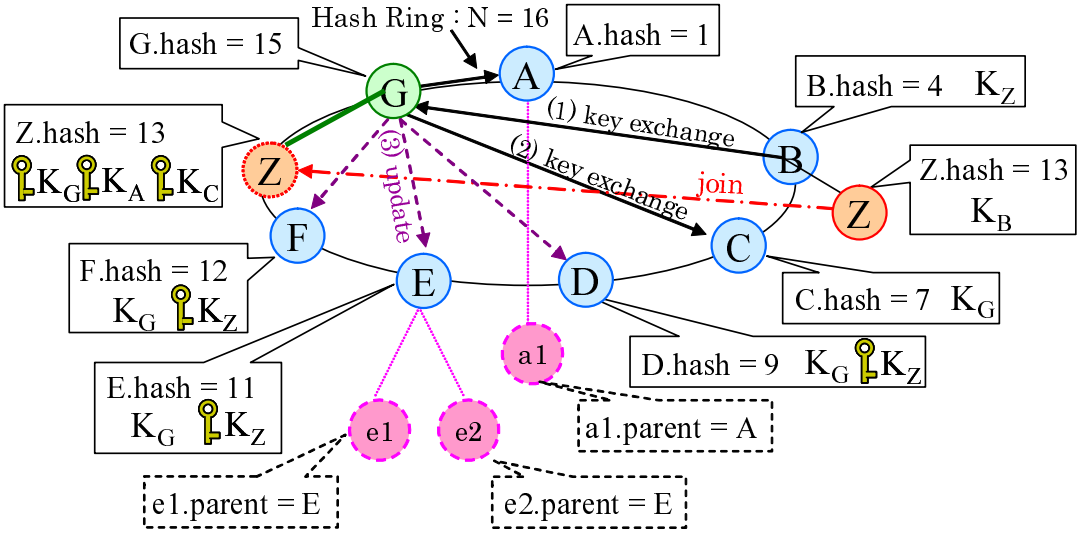

図 3.3: ノードの参加手順

3.6 ノードの離脱手順

親ノードがネットワークから離脱する場合，離脱ノードはハッシュリング上で隣接するノードに離脱通知を行い，この通知に従って隣接ノードは離脱ノードが存在しない信頼の輪を再構築する．このとき，離脱ノードに属していた子ノードは，隣接ノードの子ノードとして HiHDAM に参加し直す．また，予告なく親ノードが離脱した場合，隣接ノードがノードの離脱を検知し，隣接ノードは離脱ノードが存在しない信頼の輪を再構築し，離脱ノードの子ノードを自身の子ノードとして管理する．ネットワークに参加するノード数が n のとき，親ノードの離脱に必要な通信データ量は $O(\log n)$ となる．

子ノードがネットワークから離脱する場合，自身の親ノードに離脱を通知することで，ネットワークから離脱する．離脱通知を受信した親ノードは，その子ノードの公開鍵を破棄する．予告なく子ノードが離脱した場合には，親ノードが離脱を検知し，その子ノードの公開鍵を破棄する．

図 3.4 に親ノードが HiHDAM から離脱する際の例を示す．図 3.4 では，ノード Z がネッ

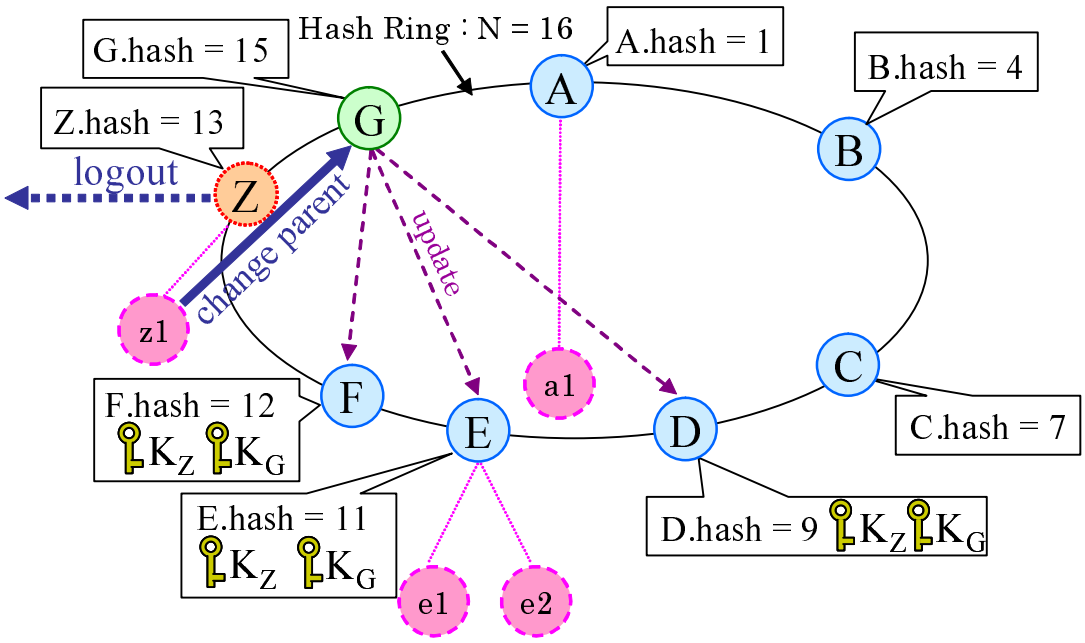

図 3.4: ノードの離脱手順

トワークから離脱している．このとき，ノード Z はまず正側に隣接しているノードである G に対して離脱を通知する．離脱通知を受けたノード G は離脱ノードであるノード G が存在しない信頼の輪の再構築を行う．具体的には，ノード D，E，F に対して，自身の公開鍵を送信する．また，ノード G はノード Z の子ノードであったノード z1 の親ノードとなる．以上のようにして，親ノード Z はネットワークから離脱する．

3.7 公開鍵の更新

信頼の輪を安全に運用するためには一定期間ごとに公開鍵の更新を行うことが必要となる．提案手法においては，各ノードが自身の古い公開鍵を保持しているノードに対して，新しい公開鍵を送信することにより，公開鍵の更新を行う．ネットワークに参加するノード数が n のとき，親ノードの公開鍵の更新に必要な通信データ量は $O(\log n)$ となる．ま

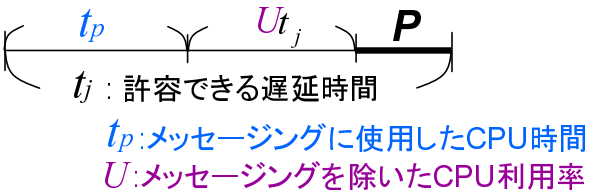

図 3.5: メッセージング性能 P

た，子ノードが公開鍵を更新する際には，自身の親ノードに対してのみ新しい公開鍵を送信することとなる．

3.8 認証のための資源状況の判定機能

提案手法においては，ネットワークに参加する各ノードをその CPU 性能などの資源状況に応じて，親ノードと子ノードに動的に分類し，各ノードの性能を考慮したオーバーレイネットワークを構築する．ここで，各ノードの資源状況を測定する指標として，あるノードのメッセージング性能 P を以下のように定義する．ここで，t_j は端末が一つのメッセージを処理する際に発生する時間における許容できる遅延時間，t_p はあるノードが一つのメッセージを処理する際に使用した CPU 時間，U はあるノードが一つのメッセージングを処理する際の，メッセージングを除いた CPU 利用率である．導出式におけるそれぞれの変数の意味を示したものを図 3.5 に示す．

$$P = t_j - Ut_j - t_p$$

この導出方法により，P の値が大きいほど，資源状況に余裕があるノードであると判定することが可能になる．加えて，P の値は各端末が自律的に導出することが可能であり，集中型のサーバなどを用いることなく，各端末の資源状況に応じてオーバーレイネットワークを階層化することが可能となる．

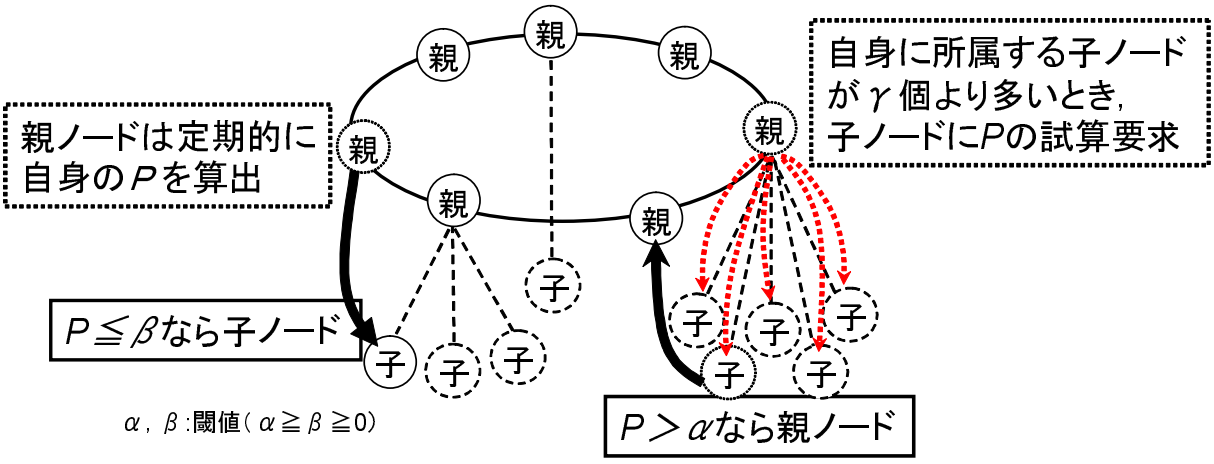

図 3.6: オーバーレイネットワーク構成法

ここで，親ノードとして充分なメッセージング性能を示す定数 α と β を用意しておき $(\alpha > \beta)$，あるノードのメッセージング性能について，$P > \alpha$ を満たす場合は親ノードとして充分な計算機資源状況であり，$P \leqq \beta$ となった場合は，親ノードとしては不充分な資源状況であると判断する．

各ノードは HiHDAM に参加する際，自身の P を算出し，$P > \alpha$ を満たしている場合は親ノードとして，それ以外の場合は子ノードとして HiHDAM に参加することとする．これにより，計算能力に優れるノードを親ノード，そうでないノードを子ノードとしてオーバーレイネットワークを構築することが可能となる．

図 3.6 に，オーバーレイネットワーク構成法について表した図を示す．ネットワークに参加している親ノードは定期的に自身の P を算出し，$P \leqq \beta$ となった場合，メッセージングに負荷が掛かりすぎていると判断し，子ノードとして HiHDAM に参加し直すこととする．さらに，親ノードは自身に所属する子ノードの台数が γ 台より大きいとき，自身に所属する子ノードすべてに対して定期的に P の試算要求を行う．そのとき，$P > \alpha$ を満たす子ノードが存在する場合には，その子ノードは親ノードとして HiHDAM に参加し直す．ここで，γ を親ノード 1 台あたりが管理する子ノードの上限値の目安として設定す

ることで，ネットワーク全体の親ノードと子ノードの数の割合を調整することが可能となる．これにより，各ノードがネットワーク参加後にCPU使用状況などに大きな変化があった場合などでも，動的にオーバーレイネットワークを構築できることで，常に目的に適したオーバーレイネットワークを構築することが可能となる．

3.9 マルチパスを用いた公開鍵の検索機能

提案手法においては，信頼の輪を用いることで，ハッシュリング上に配置された複数のノードと信頼関係を構築し，目的の公開鍵の取得を行う．しかし，公開鍵の中継が行われる際に，参加ノード中に悪意のあるノードが紛れ込んでいた場合，正しく中継が行われない危険性が生じる．そこで，提案手法においては，公開鍵を取得する際に，複数の経路（マルチパス）を用いることで，正しい公開鍵を得られる可能性を向上させる工夫を施す．

図3.7に，親ノードGが親ノードAに認証要求を行った際の，ノードAがノードGの公開鍵を取得する手続きにおいて，マルチパスを3つ用いた場合の例を示す．これは，ノードAがK_B，K_C，K_Dの3つの公開鍵を所持している状態で，ノードGの公開鍵K_Gを取得する手続きを示したものである．具体的な手順は次のようになる．

1本目のパス

1. ノードAは，自身が公開鍵を保持しているノードの中から，ノードGから逆時計周りに数えて，ハッシュリング上で最もノードGに近いハッシュ値を持つノードDに対し，Gの公開鍵K_Gを要求する．

2. ノードDは，ノードGの公開鍵K_Gを保持しているので，ノードAにK_Gを送信する．

3. ノードAはK_Gを取得する．

2本目のパス

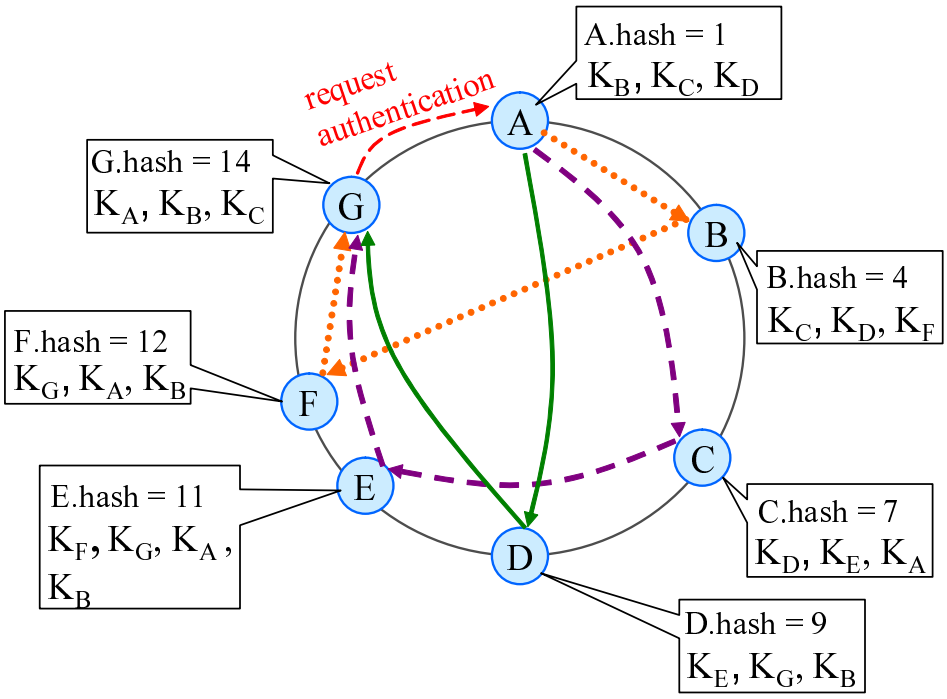

図 3.7: 親ノードによるマルチパスを用いた公開鍵取得の手順

1. ノード A は，自身が公開鍵を保持しているノードの中から，ノード G から逆時計周りに数えて，ハッシュリング上で 2 番目にノード G に近いハッシュ値を持つノード C に対し，G の公開鍵 K_G を要求する．

2. ノード C は，ノード G の公開鍵 K_G を保持していないので，C が持つ公開鍵のうち，ノード G から逆時計周りに数えて最もノード G に近いハッシュ値を持つノード E の公開鍵 K_E を送信する．これにより，ノード A は K_E を取得し，ノード E と安全な通信を行うことが可能となる．

3. ノード A はノード E に対して，ノード G の公開鍵 K_G を要求する．

4. ノード E は，ノード G の公開鍵 K_G を保持しているので，ノード A に K_G を送信する．

5. ノード A は K_G を取得する．

3 本目のパス

1. ノード A は，自身が公開鍵を保持しているノードの中から，ノード G から逆時計周りに数えて，ハッシュリング上で 3 番目にノード G に近いハッシュ値を持つノード B に対し，G の公開鍵 K_G を要求する．

2. ノード B は，ノード G の公開鍵 K_G を保持していないので，B が持つ公開鍵のうち，ノード G から逆時計周りに数えて最もノード G に近いハッシュ値を持つノード F の公開鍵 K_F を送信する．これにより，ノード A は K_F を取得し，ノード F と安全な通信を行うことが可能となる．

3. ノード A はノード F に対して，ノード G の公開鍵 K_G を要求する．

4. ノード F は，ノード G の公開鍵 K_G を保持しているので，ノード A に K_G を送信する．

5. ノード A は K_G を取得する．

以上のように，ノード A は 3 種類の経路を用いてノード G の公開鍵 K_G の取得を試みることで，より安全で確実な検索を行うことが可能となる．

第4章　評価

本章では，計算機シミュレーションによる実験を通して，提案手法の有効性を評価する．まず，計算機シミュレーションにより行った実験の環境や条件について述べ，本章で行う評価の項目について整理する．次に，実験結果から提案手法が評価項目で挙げられた条件を満たしているかを確認し，最後に，考察を行う．

4.1　実験概要

ここでは，実験の条件について詳細に説明する．本節では，シミュレータの目的と環境について説明した後，実験で想定したシナリオについて詳細に説明する．

4.1.1　シミュレータ概要

本実験では，Java により実装した独自のシミュレータを用いた．このシミュレータでは，全てのノードが相互通信可能な状態を想定しており，ネットワークの障害や遅延などは考慮していない．その上で，提案手法による認証ネットワークの構成・公開鍵の配布のみを行い，ネットワークに参加する各ノードの状態や，所持している公開鍵数，送受信された通信メッセージ数などを測定する．図 4.1 に，今回用いたシミュレータにおける物理的なネットワーク構成のモデルを示す．

ネットワークに参加するすべてのノードはエージェント（ノードエージェント）として実現されており，ノードエージェント間でメッセージを送受信することにより，ネットワークへの参加時，ネットワークからの離脱時，公開鍵の更新時，及び公開鍵の取得要求が行われた際のノード間の通信をシミュレートする．図 4.2 にノードエージェントの状態

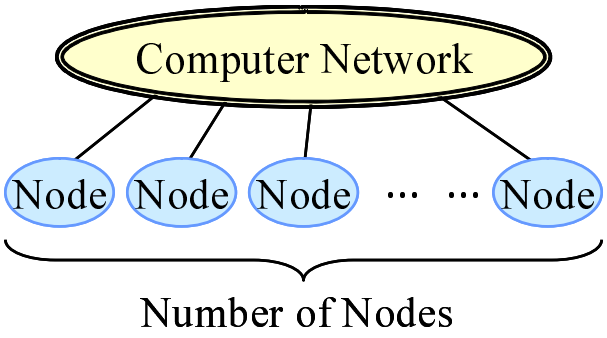

図 4.1: シミュレータにおける物理的なネットワーク構成のモデル

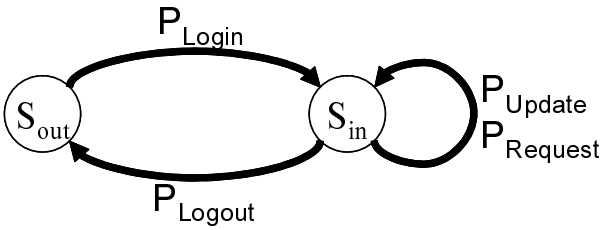

図 4.2: ノードエージェントの状態遷移

遷移図を示す．ノードエージェントはネットワークに参加していない状態 (S_{out}) とネットワークに参加している状態 (S_{in}) を持つ．ノードエージェントは状態 S_{out} の時に確率 P_{login} でネットワークに参加し，状態 S_{in} に移行する．同様に状態 S_{in} の時は確率 P_{logout} でネットワークから離脱し，状態 S_{out} に移行する．また，状態 S_{in} の時は確率 P_{update} で自身の公開鍵を更新し，確率 $P_{request}$ で乱数に従って決定された受信ノードに対して公開鍵を要求するメッセージを送信する．

4.1.2 実験パターン

実験において，ノードエージェントの行動パターンとして，表4.1に示した2つのパターンを用意した．Pattern 1はノードがネットワークへの参加と離脱を頻繁に繰り返す場面を想定しており，Pattern 2はネットワークに参加したノードの離脱があまり行われない場面を想定している．

本研究においては，様々な端末がネットワークに参加や離脱を頻繁に繰り返すことが予想されるユビキタス情報環境への適用を目的としているため，Pattern 1における結果が特に重要となる．

4.1.3 シナリオ設定

今回の実験においては，CPU性能の高いノードと低いノードがそれぞれ参加する環境を想定するため，ノードの種類としてA,B,Cの3つを用意し，メッセージング能力t_pをそれぞれ設定した．ノードAは性能の高いサーバなどといった計算端末を想定し$t_p = 600ms$，ノードBは一般的な据え置き端末を想定し$t_p = 1000ms$，また，ノードCは携帯電話などの携帯端末を想定し$t_p = 2000ms$とした．各ノードの台数の割合は，ノードAがノード数全体の10%，ノードBが20%，ノードCが70%であるものとした．これは，ユビ

表 4.1: 実験パターン

Pattern	P_{login}	P_{logout}	P_{update}	$P_{request}$
no. 1 **（参加・離脱が頻繁）**	1.0	**0.45**	0.05	0.5
no. 2 （参加・離脱が少ない）	1.0	0.01	0.01	0.98

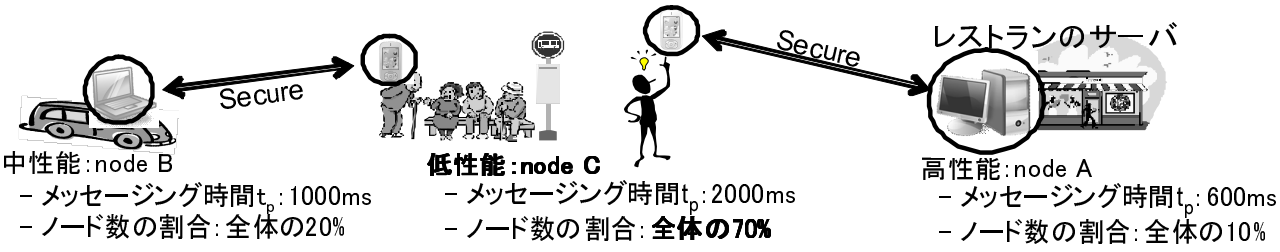

図 4.3: 実験シナリオ

キタス情報環境においては，ネットワークに参加する端末のうち，携帯端末やセンサデバイスなどの比較的低性能な端末が占める割合が大きくなることが予測されるためである．想定したシナリオを図 4.3 に示す．また，各ノードの CPU 使用率 U に関しては，0% を最も発生し易い頻度としたべき分布に従う乱数によって算出するものとした．提案手法における閾値については主に，$t_j = 2000ms, \alpha = 500, \beta = 0ms, \gamma = 2$ 台 とした．基本的な実験条件をまとめたものを表 4.2 に示す．本論文中で示した実験のうち，特に条件を明示していない実験は，表 4.2 に示した実験条件で行ったものである．

表 4.2: 実験条件

項目	値
ノード数	10 〜 1000 台
N（最大ハッシュ値）	1024
t_j（許容できる遅延値）	2000ms
α（親ノードとなる閾値）	500ms
β（子ノードとなる閾値）	0ms
γ（試算要求を行う閾値）	2 台
U（各ノードの CPU 使用率）	べき分布に従いランダムに算出

4.2 評価項目

本章では，提案手法の有効性の評価として，計算機シミュレーションを用いて以下の3項目について評価を行う．

（性能評価）提案手法が低性能端末でも参加可能なスケーラブルな分散認証手法であることを確認する．

（機能確認）提案手法が端末の資源状況に応じて，認証のためのオーバーレイネットワークを動的に構築出来ているか確認する．

（耐障害性評価）提案手法の耐障害性を検証する．

（性能評価）では，ネットワークに参加するノード間でやりとりされた通信メッセージ数を計測することで，低性能端末に必要なメッセージの処理負荷が削減できていることを確認する．また，ネットワークに参加する各ノードが保持している公開鍵の個数を計測することで，低性能端末に必要となるメモリ量が増加していないことを確認する．

（機能確認）では，ネットワークに参加するノードの種別（親ノード・子ノード）の割合を計測していくことで，提案手法において，資源状況に余裕がなくなったノードが動的に子ノードとなり，負担を軽減できているかを確認する．また，親ノード1台あたりに生じるメッセージ処理負荷について，単純に一部の高性能端末のみを親ノードとした場合と比較を行い，提案手法を用いることで，一部の高性能端末のみに負荷が集中していないことを確認する．加えて，提案手法において用いる閾値を変化させた場合に，構築されるオーバーレイネットワークにどのような影響が出るのか分析する．

（耐障害性評価）では，認証ネットワーク中に悪意のあるノードが混在する状態において公開鍵の検索の成功率を計測することで，提案手法の耐障害性を検証する．その際，マルチパスを用いて公開鍵の検索を行うことで，公開鍵取得の成功率が向上することを確認する．また，マルチパスを用いた場合における各ノードのメッセージ処理にかかる負荷を計測することで，提案手法がマルチパスを用いる場合においても，各端末に生じる負荷が極端に増加せず，スケーラビリティに優れていることを確認する．

4.3　実験結果

本節では，シミュレーション実験から得られた結果を示し，提案手法の有用性を評価する．

4.3.1　性能評価

ここでは，提案手法が低性能端末でも参加可能なスケーラブルな分散認証手法であることを確認するため，各端末に必要となる通信メッセージ量とメモリ量について評価を行う．今回，比較として用いた既存手法は HDAM[26] である．

各端末にかかる処理負荷

まず，各ノードに生じる処理負荷を測定するために，各ノードが認証のために送受信した通信メッセージ数の計測を行った．認証のために送受信されるメッセージには，公開鍵暗号方式による暗号化や署名が施してあるため，メッセージを受信した端末はその解読のための計算を行うことが強いられる．認証ネットワークに参加するノード数を 10 台から 1000 台まで増加させていき，すべてのノードが図 4.2 に示した状態遷移を 10 回行ったとき，各ノード 1 台あたりが 1step 中に受信したメッセージ数の平均の個数について，表 4.1 に示した Pattern 1 と Pattern 2 のそれぞれの場合について計測したものを図 4.4，図 4.5 にそれぞれ示す．ここで，1step とは，すべてのノードが状態遷移を 1 回ずつ行ったことを示す単位とする．

既存手法の場合は，ネットワークに参加するノード数が増加するに従い，すべてのノードにかかる負荷が徐々に増加していくことが分かる．これは，Pattern 1 の場合において特に顕著であることから，既存手法はノードの参加と離脱が頻繁に繰り返されるユビキタス情報環境には適していないということが分かる．一方，提案手法においては，低性能端末に生じる負担が，ネットワークに参加するノードの総数に関わらず常にほぼ一定である．よって，提案手法は多数のノードがネットワークに頻繁に参加や離脱を繰り返すユビキタ

ス情報環境において特に有効であることが分かる．また，提案手法における高性能端末に生じる負荷も，既存手法のすべてのノードに生じる負荷と比較し，極めて少量の増加に留まっており，提案手法は極めてスケーラブルな分散認証手法であるといえる．

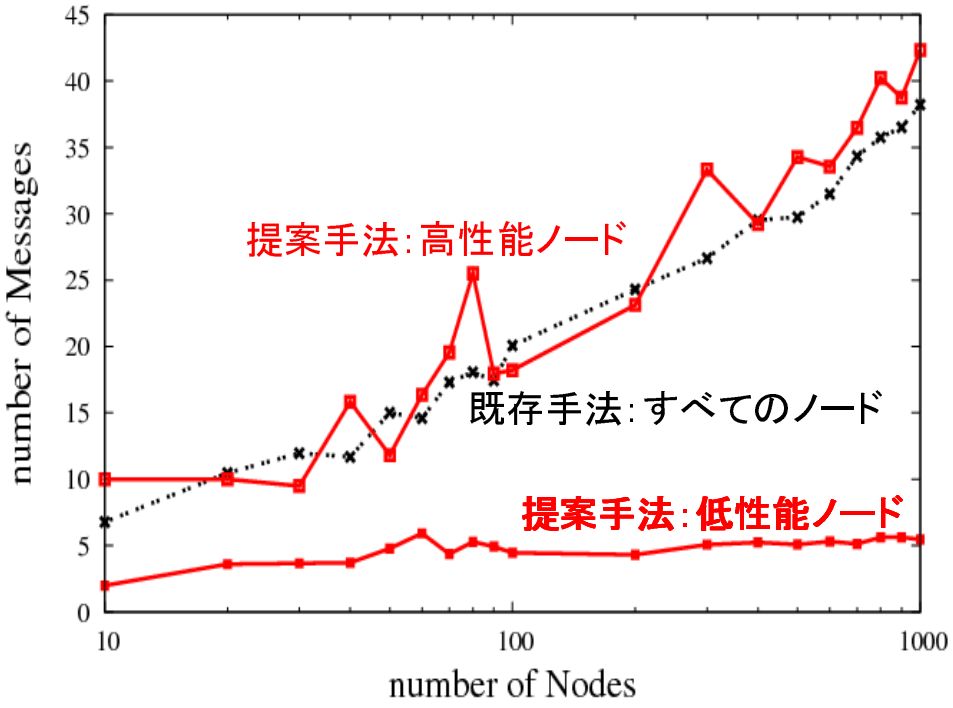

図 4.4: Pattern 1 における各ノードの処理メッセージ数

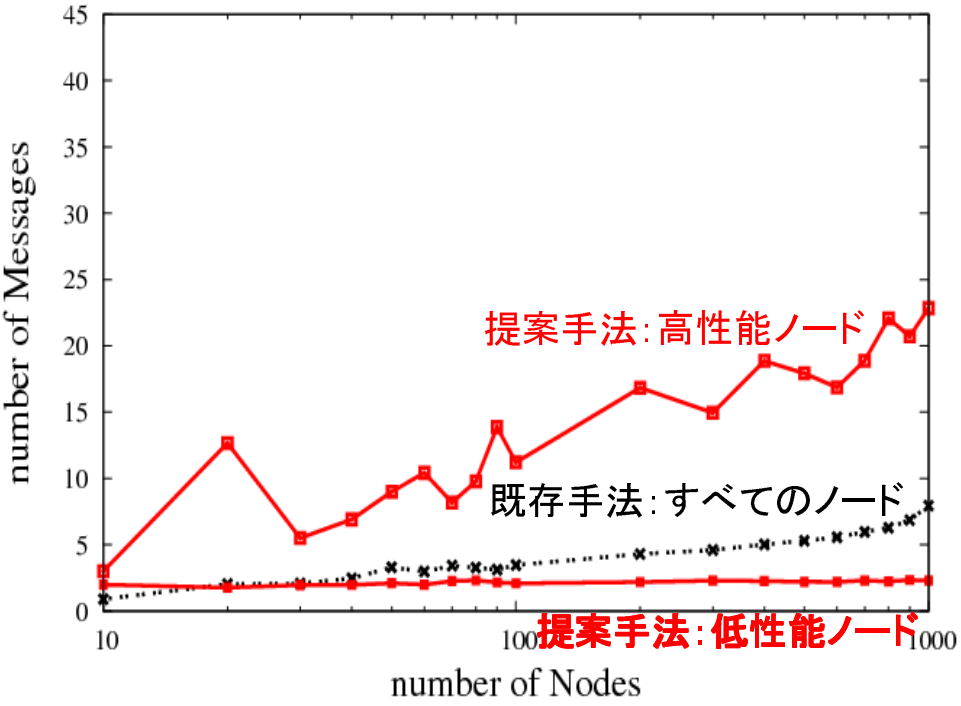

図 4.5: Pattern 2 における各ノードの処理メッセージ数

必要なメモリ量

次に，各ノードに必要となるメモリ量を測定するために，各ノードが認証のために管理していた公開鍵の個数を計測した．さきほどと同様に，認証ネットワークに参加するノード数を 10 台から 1000 台まで増加させていき，すべてのノードが図 4.2 に示した状態遷移を 10 回行ったとき，各ノード 1 台あたりが管理していた公開鍵の個数の平均について，表 4.1 に示した Pattern 1 と Pattern 2 の両方についてそれぞれ計測したものを図 4.6, 図 4.7 にそれぞれ示す．

Pattern 1，Pattern 2 いずれの場合においても，提案手法における低性能端末に必要となるメモリ量は常に一定であることが分かる．また，既存手法と比較して，必要となるメモリ量が極端に増加してしまうこともなく，その必要なメモリ量は高性能端末に $O(\log n)$（n：参加ノード数）である．よって提案手法は，低性能端末でも参加可能なスケーラブルな分散認証手法であるといえる．

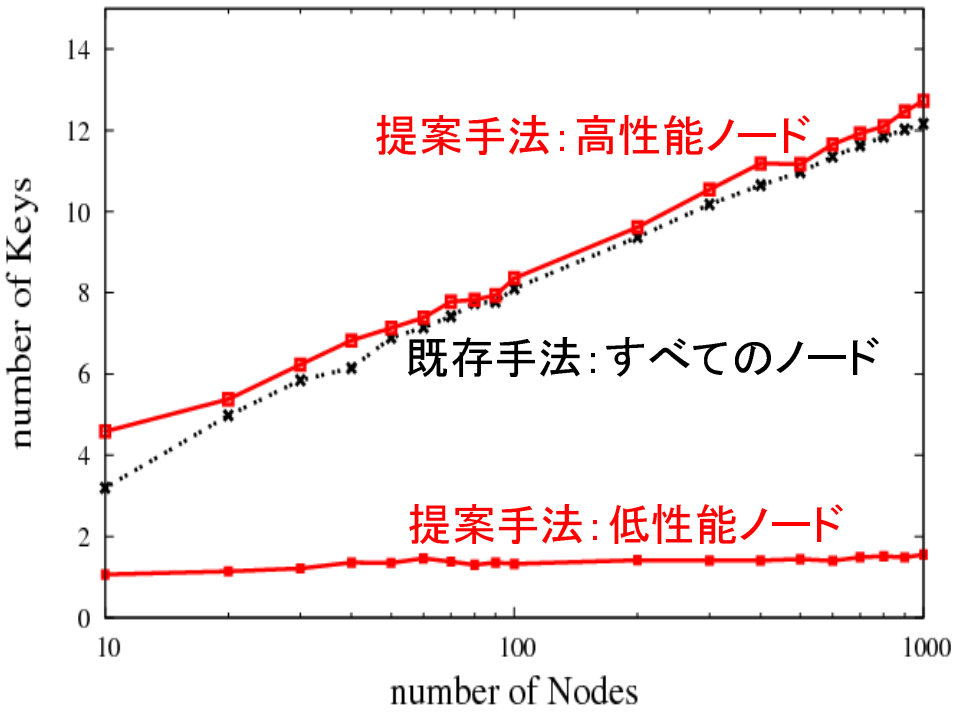

図 4.6: Pattern 1 における各ノードの管理する公開鍵数

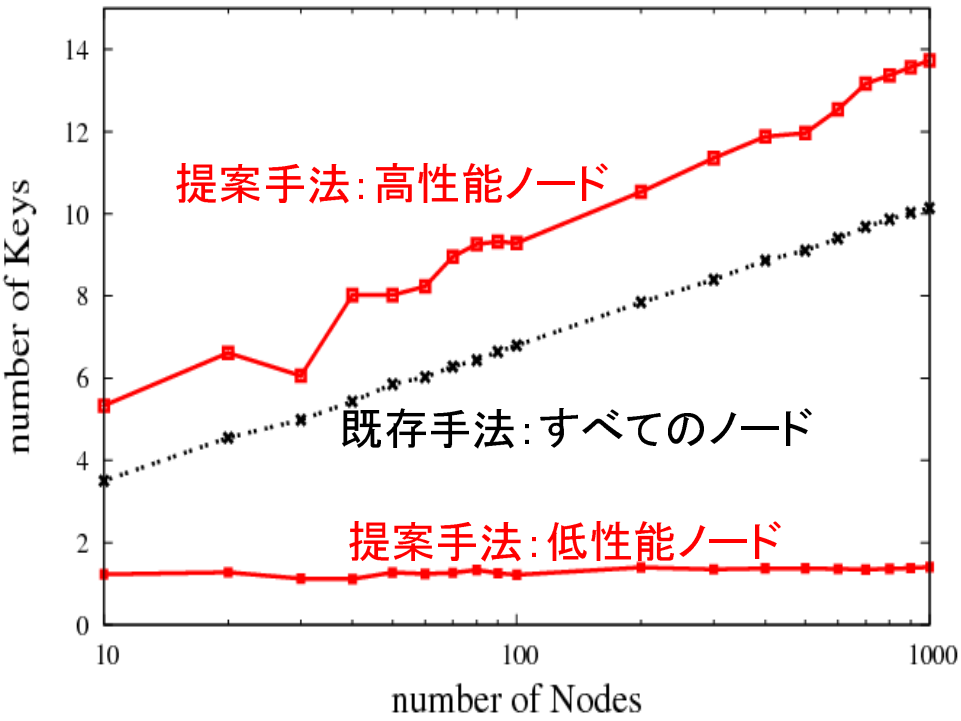

図 4.7: Pattern 2 における各ノードの管理する公開鍵数

4.3.2 機能確認

ここでは，提案手法が端末の資源状況に応じて，認証のためのオーバーレイネットワークを動的に構築可能であるかどうかを確認するため，ネットワークに参加するノードの種別の割合について確認を行う．また，提案手法を適用した場合と，単純に高性能端末だけに認証のための処理負荷を集中させた場合とを比較し，各端末に生じる負荷についてどのような違いが出るのかについても評価を行う．

オーバーレイネットワークの構成確認

提案手法を適用した際，各端末の資源状況に応じて，動的にオーバーレイネットワークが構成されているかどうかを確認するため，親ノードと子ノードがどのような割合で存在していたかを計測する．

表 4.1 で示した pattern 1 の場合で実験を行いノード数が 1000 台に達した際に，図 4.3 で示した 3 種類のノードについて，それぞれがどのような割合で親ノードと子ノードに分類されていたかを表 4.3 に示す．なお，ここで示したノードの種別の割合に関しては，参加ノード数の大小に関わらず，常にほぼ一定であった．CPU 性能の低いノード C については，親ノードとなるための条件が不充分であるため，確実に子ノードとなっていることがわかる．また，CPU 性能の高いノード A に関しては，9 割程度のノードが親ノードとなり，1 割程度は子ノードとなっていることがわかる．これは，CPU 性能が優れている計算機端末の場合でも，アプリケーションの使用状況などに応じて CPU 利用率が高まり，メッセージングを行う余裕がなくなってきたと判断された場合は，子ノードとして HiHDAM に参加するためである．よって，提案手法は各ノードの持つ資源に応じて動的にオーバーレイネットワークを構築可能な手法であるといえる．

親ノードが処理したメッセージ数

図 4.8 に，提案手法における親ノードと子ノードの動的な調整を行った場合と，表 4.3 におけるノード A のみを親ノードとした場合とについて，ネットワークに参加する全て

のノード数と，親ノード 1 台あたりが処理するメッセージ数の関係を示す．これは，認証ネットワークに参加するノード数を 10 台から 1000 台まで増加させていき，すべてのノードが表 4.1 に示した Pattern 1 の確率に従い，図 4.2 に示した状態遷移を 10 回行ったとき，各ノード 1 台あたりが 1step（すべてのノードに状態遷移を 1 回ずつ行わせる）あたりに受信したメッセージ数の平均の個数について計測を行ったものである．

図 4.8 より，提案手法を用いることで，単純に高性能ノードのみを親ノードとする場合に比べ，親ノード 1 台あたりに生じる負荷を削減可能であることが分かる．このことは，ネットワークへの参加ノード数が大きいほど効果が高いことが分かる．これは，提案手法により，資源状況に応じて親ノードと子ノードの調整を行った場合は，表 4.3 におけるノード B の中の資源状況に余裕のあるノードも親ノードとしてネットワークに参加するため，参加ノード全体のおよそ 25% 程度が親ノードとして扱われることとなり，親ノード 1 台あたりの負荷が削減されているためである．これに対し，動的な調整を行わずノード A のみを親ノードとした場合は親ノードがノード全体の 10% しか存在しないため，性能の高いノード A すべてに大きな負荷が生じてしまうことがわかる．また，提案手法においては CPU 性能の高いノードであってもメッセージングを行うための資源状況に余裕がなくなってきた場合，自動的に子ノードとしてネットワークに参加することになるため，より安定した公開鍵管理を実現することが可能となる．

表 4.3: ノードの性能ごとの親ノードと子ノードの数の割合

Node name	親ノード	子ノード	メッセージング時間	存在割合
node A	92%	8%	600ms	10%
node B	82%	18%	1000ms	20%
node C	0%	100%	2000ms	70%

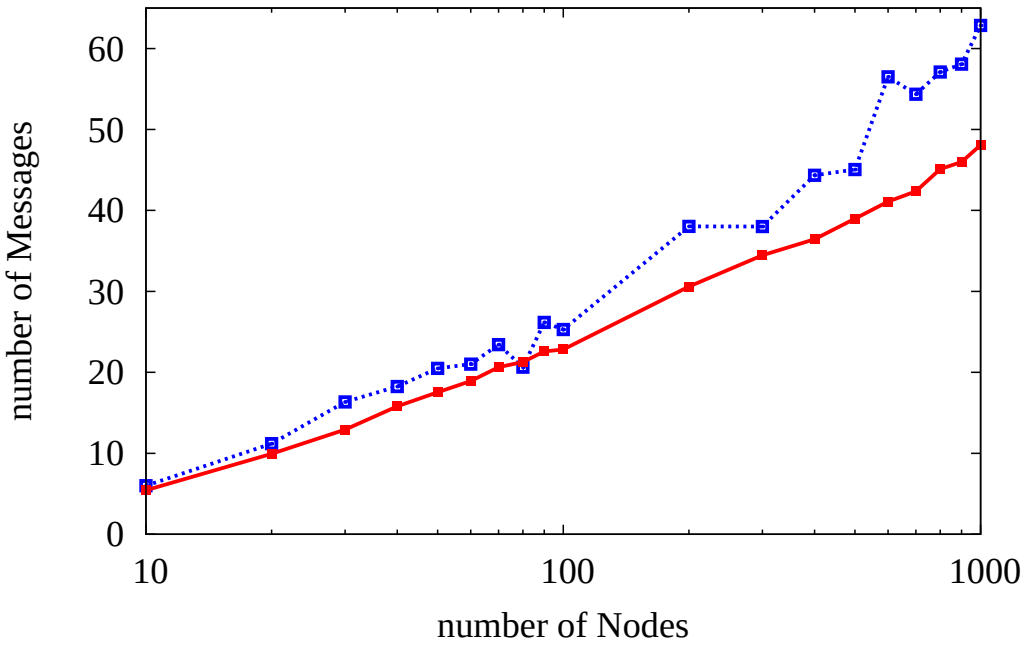

図 4.8: 親ノード 1 台あたりが処理したメッセージ数

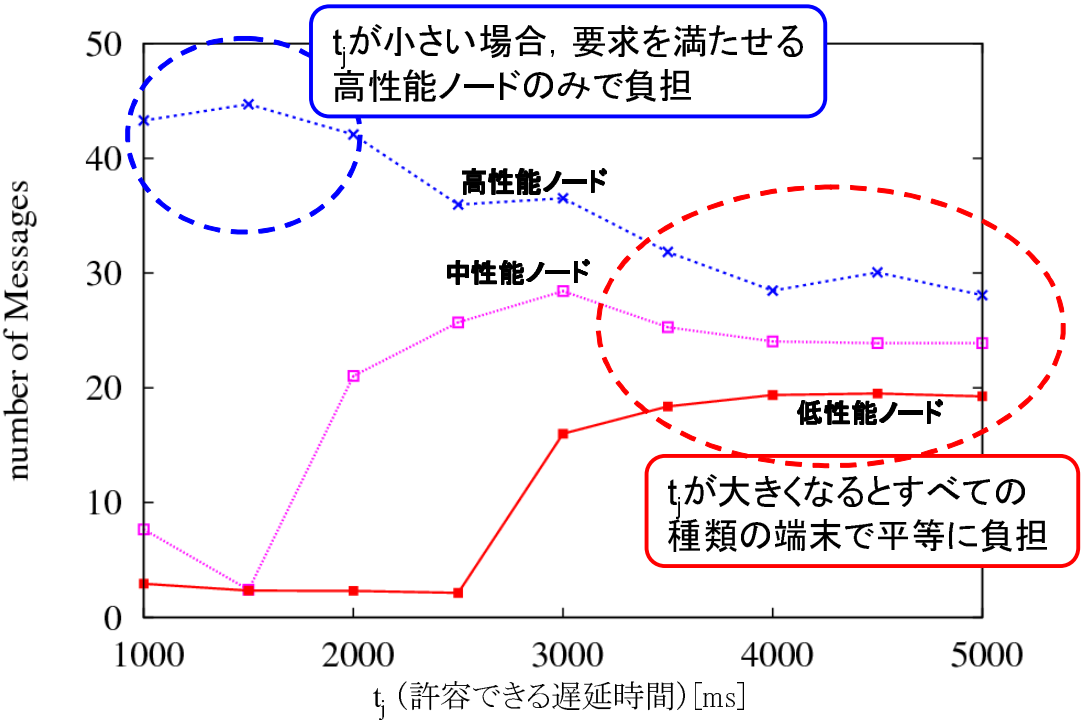

図 4.9: 性能別ノードごとの 1 ノードあたりの処理メッセージ数

許容できる遅延 t_j を変化させた場合の分析

ここでは，提案手法における閾値 t_j（端末が 1 つのメッセージを処理するのに許容できる遅延時間）の値を変化させた場合，構築されるオーバーレイネットワークにどのような影響が出るのか分析する．ネットワークに 1000 台の端末がいる状況で，各端末が表 4.1 で示した pattern 1 の割合で状態変化を行うとし，$\alpha = 500ms$，$\beta = 0ms$，$\gamma = 2$ 台 と固定して実験を行った．その条件下において，許容できる遅延時間である t_j を 1000ms から 5000ms まで変化させた場合，ノード 1 台あたりが処理するメッセージ量がどのように変化していくのか，図 4.3 に示した性能別ノードごとについて示したものを図 4.9 に示す．t_j が小さい場合においては，要求を満たせる高性能ノードのみで処理を負担することが多くなるため，高性能端末であるノード A に負荷が集中していることが分かる．また，t_j が増加するに従い，低性能端末であっても，親ノードとなる条件を満たすことが出来るよう

になるため，すべての種別のノードに平等に負荷が生じるようになることが分かる．よって，t_j の値を適切に定め提案手法を用いることで，ネットワークに参加する端末の負荷を最小限に抑えることが可能となり，低性能端末が多数混在した状態においてもスケーラビリティを維持することが可能となる．

子ノードから親ノードに変わる閾値 α を変化させた場合の分析

ここでは，提案手法における閾値 α（子ノードから親ノードに変わる閾値）の値を変化させた場合，構築されるオーバーレイネットワークにどのような影響が出るのか分析する．ネットワークに 1000 台の端末がいる状況で，各端末が表 4.1 で示した pattern 1 の割合で状態変化を行うとし，$t_j = 2000ms$，$\beta = 0ms$，$\gamma = 2$ 台 と固定して実験を行った．その条件下において，子ノードから親ノードに変わる閾値である α を-1000ms から 3000ms まで変化させた場合，親ノードと子ノードそれぞれについて，1 台あたりに処理するメッセージ量がどのように変化していくのかを図 4.10 に示す．α が β 未満である場合，そうでない場合に比べ親ノード 1 台あたりに生じる負荷が大きいことが分かる．これは，α が β 未満である場合には，オーバーレイネットワークの再構成が必要以上に行われてしまい，各端末に無駄な負荷が生じてしまうためであると考えられる．また，α が t_j に近づくにつれ，各親ノードに生じる負荷が大きくなり，t_j 以上になると，親ノードに生じる負荷が極端に大きくなることが分かる．これは，α の値を大きく設定しすぎてしまうと，親ノードなるための条件を満たせるノードがほとんど存在しなくなってしまい，一部の親ノードに大幅な負荷が生じてしまうからであると考えられる．閾値 α を変化させた場合のネットワークに参加するノードの親ノードと子ノードの割合の変化について，ノードの性能別に示したものを図 4.11，図 4.12，図 4.13 にそれぞれ示す．これらの分析結果から，α の値を適切に定め提案手法を用いることで，ネットワークに参加する端末の負荷を最小限に抑えることが可能となり，低性能端末が多数混在した状態においてもスケーラビリティを維持することが可能となることが分かる．

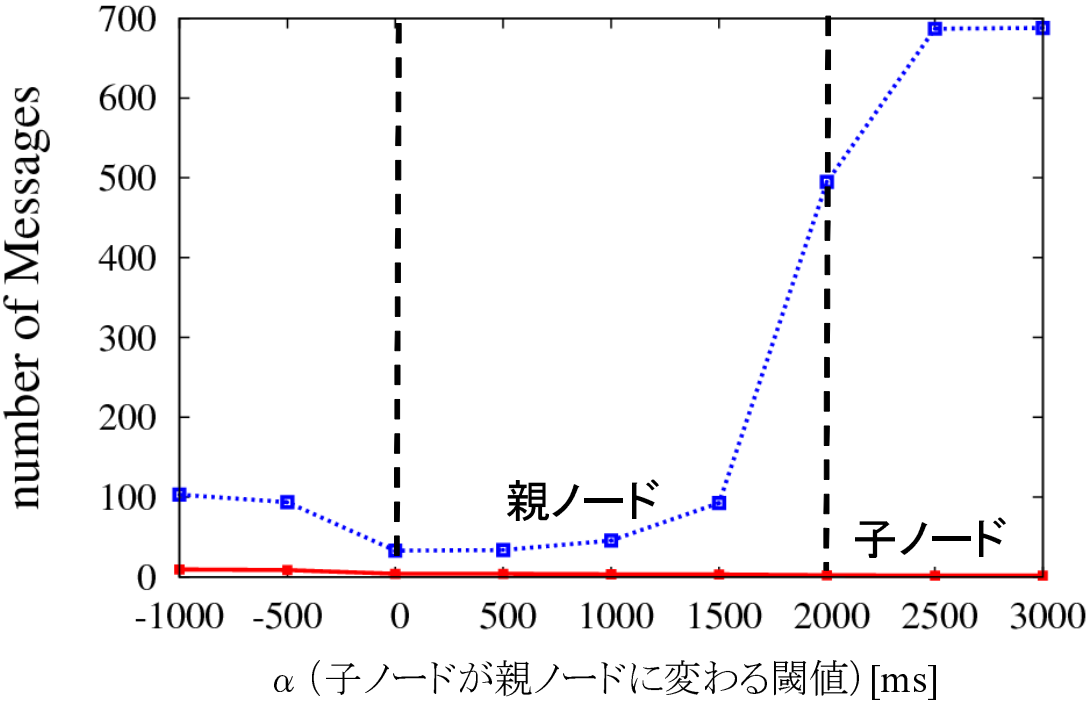

図 4.10: ノード種別ごとの 1 ノードあたりの処理メッセージ数

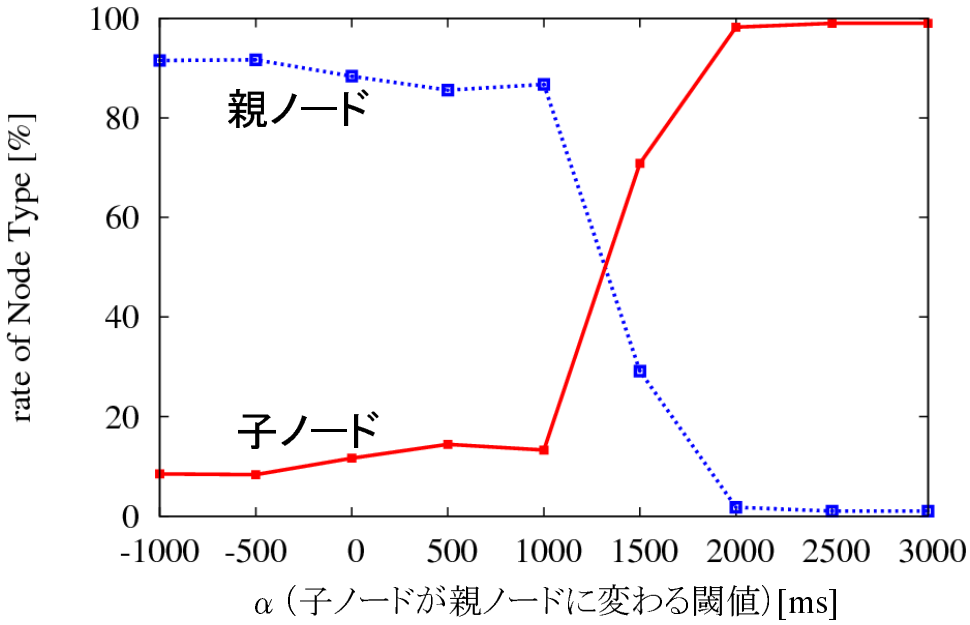

図 4.11: 高性能ノードの親ノードと子ノードの割合の変化

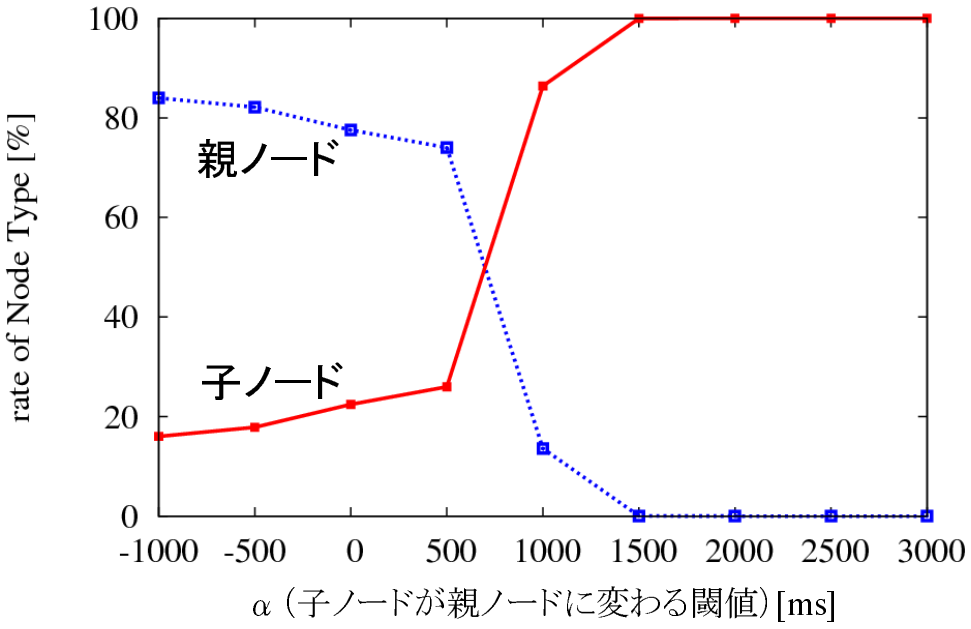

図 4.12: 中性能ノードの親ノードと子ノードの割合の変化

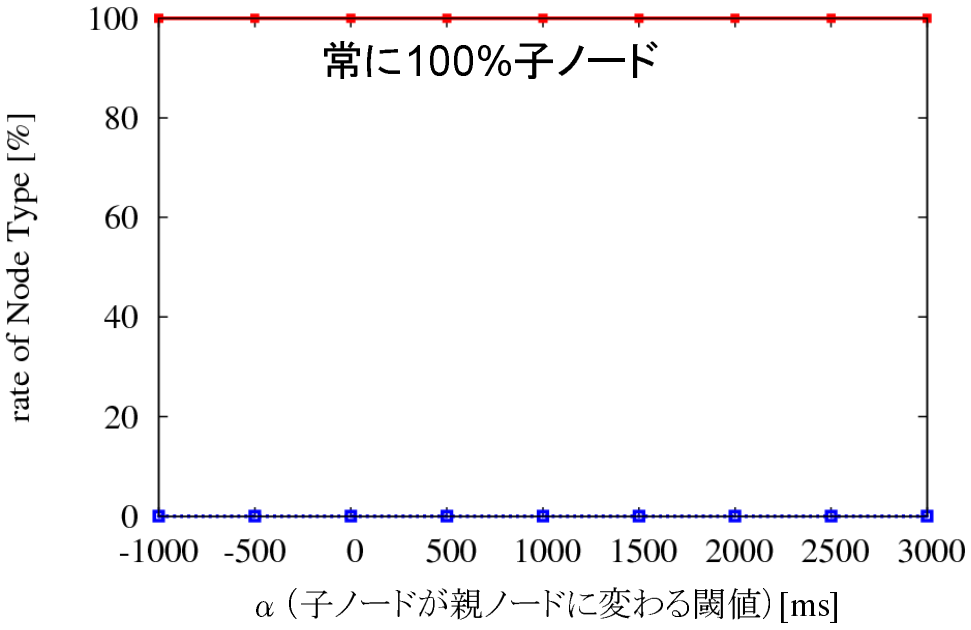

図 4.13: 低性能ノードの親ノードと子ノードの割合の変化

親ノードから子ノードに変わる閾値 β を変化させた場合の分析

ここでは，提案手法における閾値 β（親ノードから子ノードに変わる閾値）の値を変化させた場合，構築されるオーバーレイネットワークにどのような影響が出るのか分析する．ネットワークに 1000 台の端末がいる状況で，各端末が表 4.1 で示した pattern 1 の割合で状態変化を行うとし，$t_j = 2000ms$，$\alpha = 500ms$，$\gamma = 2$ 台 と固定して実験を行った．その条件下において，子ノードから親ノードに変わる閾値である β を-1000ms から 3000ms まで変化させた場合，親ノードと子ノードそれぞれについて，1 台あたりに処理するメッセージ量がどのように変化していくのかを図 4.14 に示す．β が大きくなるに従い，親ノード 1 台あたりに生じる負荷が徐々に増加することが分かる．特に，β が α よりも大きくなると，親ノードに生じる負荷が大幅に増加することに加え，子ノード 1 台あたりに生じる負荷も増加することが分かる．これは，β が大きくなるにつれて，親ノードから子ノードに変化するノードが増加することで，ネットワーク全体における親ノードが占める割合が極端に低下することで，一部の親ノードに大きな負荷が集中しているからである．さらに，$\beta > \alpha$ の場合は，子ノードが自身のメッセージング性能 P を算出し，それが α 以上のときに親ノードとしてネットワークに参加し直しても，すぐに P が β を下回ることで，一旦ネットワークから離脱し，子ノードとして参加し直すということが非常に頻繁に発生してしまう．これにより，ノード間で送受信されるメッセージ数が大幅に増加してしまい，各ノードへの負荷が非常に大きなものになってしまっていると考えられる．閾値 β を変化させた場合のネットワークに参加するノードの親ノードと子ノードの割合の変化について，ノードの性能別に示したものを図 4.15，図 4.16，図 4.17 にそれぞれ示す．これらの分析結果から，β の値を適切に定め提案手法を用いることで，ネットワークに参加する端末の負荷を最小限に抑えることが可能となり，低性能端末が多数混在した状態においてもスケーラビリティを維持することが可能となることが分かる．

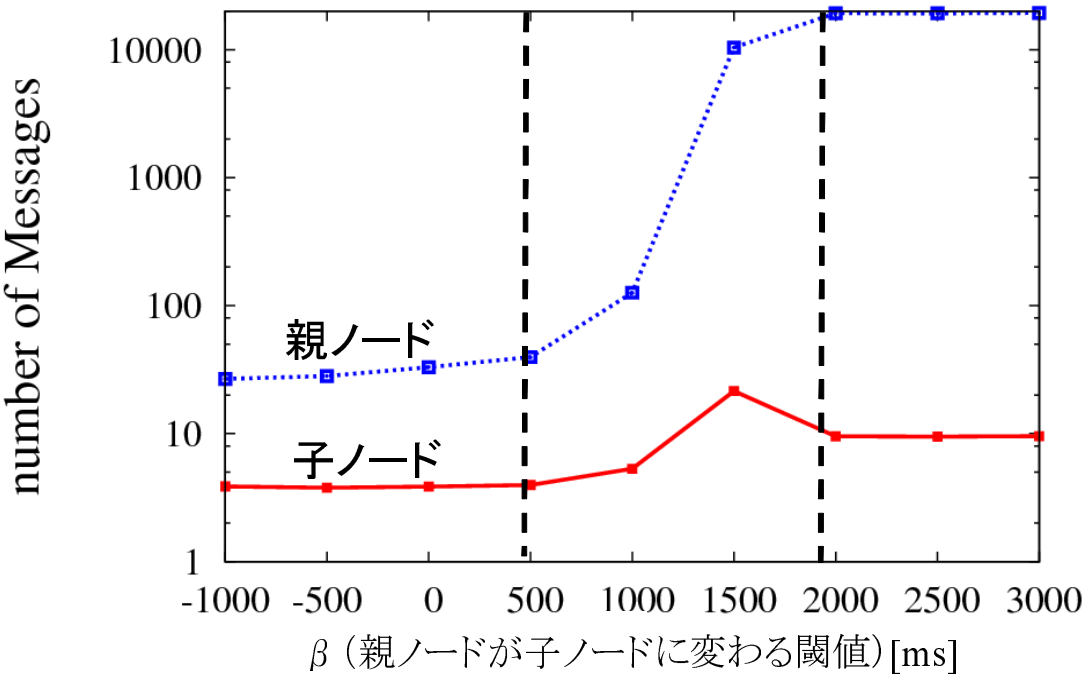

図 4.14: ノード種別ごとの 1 ノードあたりの処理メッセージ数

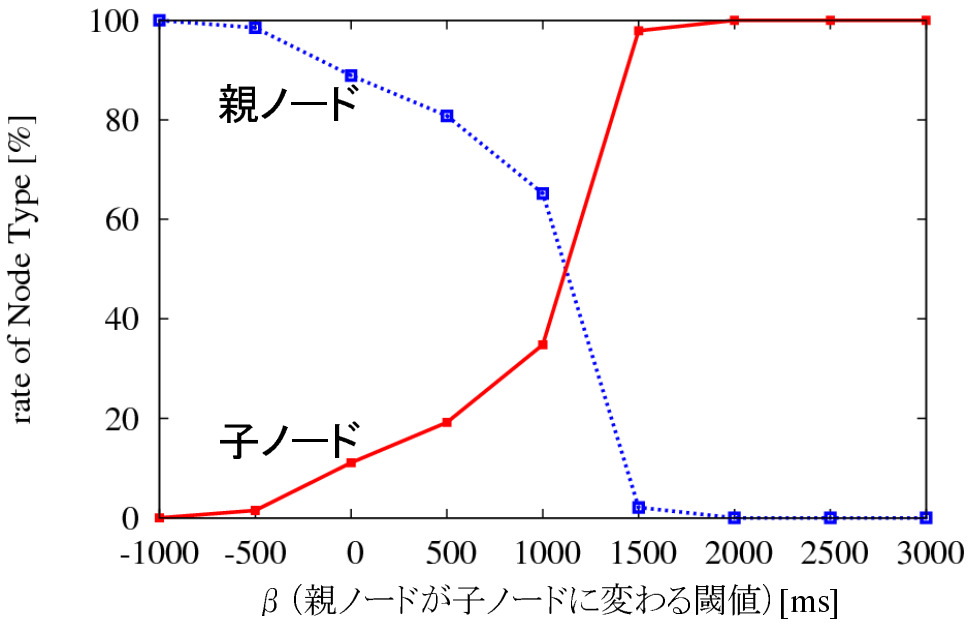

図 4.15: 高性能ノードの親ノードと子ノードの割合の変化

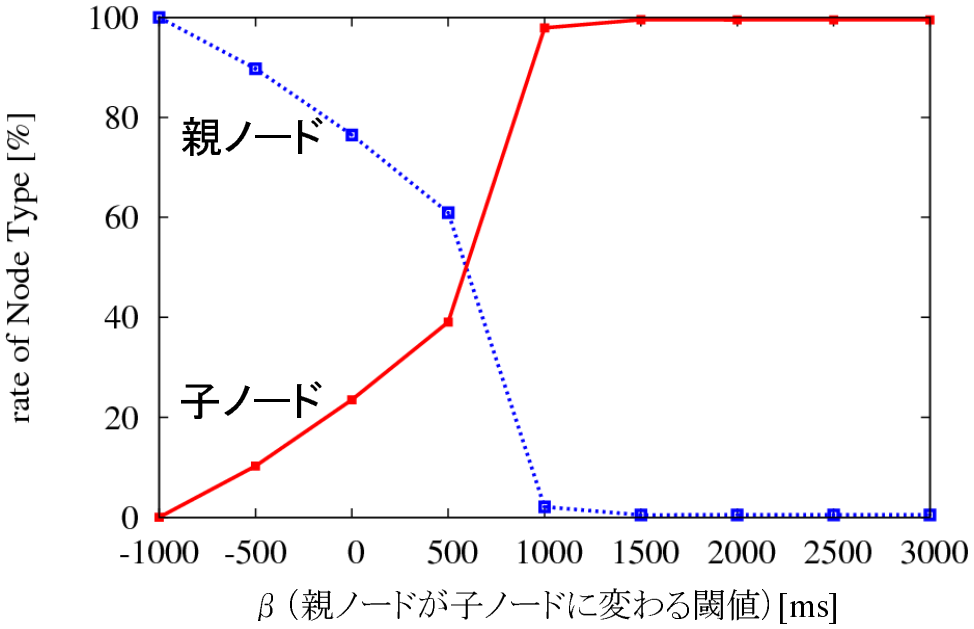

図 4.16: 中性能ノードの親ノードと子ノードの割合の変化

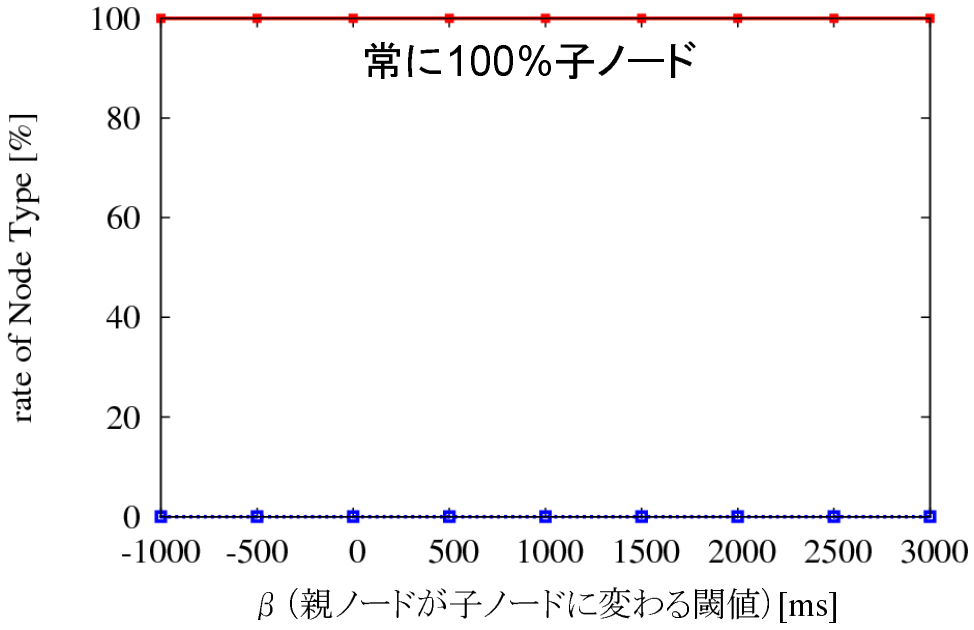

図 4.17: 低性能ノードの親ノードと子ノードの割合の変化

親ノードが子ノードに試算要求を開始する子ノード数を決定する閾値γを変化させた場合の分析

提案手法における親ノードは自身に所属する子ノードの台数がγ台より大きいとき，自身に所属する子ノードすべてに対して定期的にPの試算要求を行う．ここでは，閾値γの値を変化させた場合，構築されるオーバーレイネットワークにどのような影響が出るのか分析する．ネットワークに1000台の端末がいる状況で，各端末が表4.1で示したpattern 1の割合で状態変化を行うとし，$t_j = 2000ms$，$\alpha = 500ms$，$\beta = 0ms$と固定して実験を行った．その条件下において，閾値γを0台から10台まで変化させた場合，親ノードと子ノードそれぞれについて，1台あたりが処理するメッセージ量がどのように変化していくのかを図4.18に示す．γの値が大きくなることで，親ノード1台あたりに生じる負荷がゆるやかに増加傾向を示すことが分かる．これは，γの値が大きくなることで，親ノードが子ノードに対して試算要求を行う機会が減り，子ノードから親ノードに変化するノードが少なくなるため，ネットワーク全体に存在する親ノードの台数が減少し，親ノード1台あたりに生じる負荷が増加するためであると考えられる．また，子ノードについてはγの値に関わらず，常に極めて少量の負荷であることが分かる．閾値γを変化させた場合のネットワークに参加するノードの親ノードと子ノードの割合の変化について，ノードの性能別に示したものを図4.19，図4.20，図4.21にそれぞれ示す．

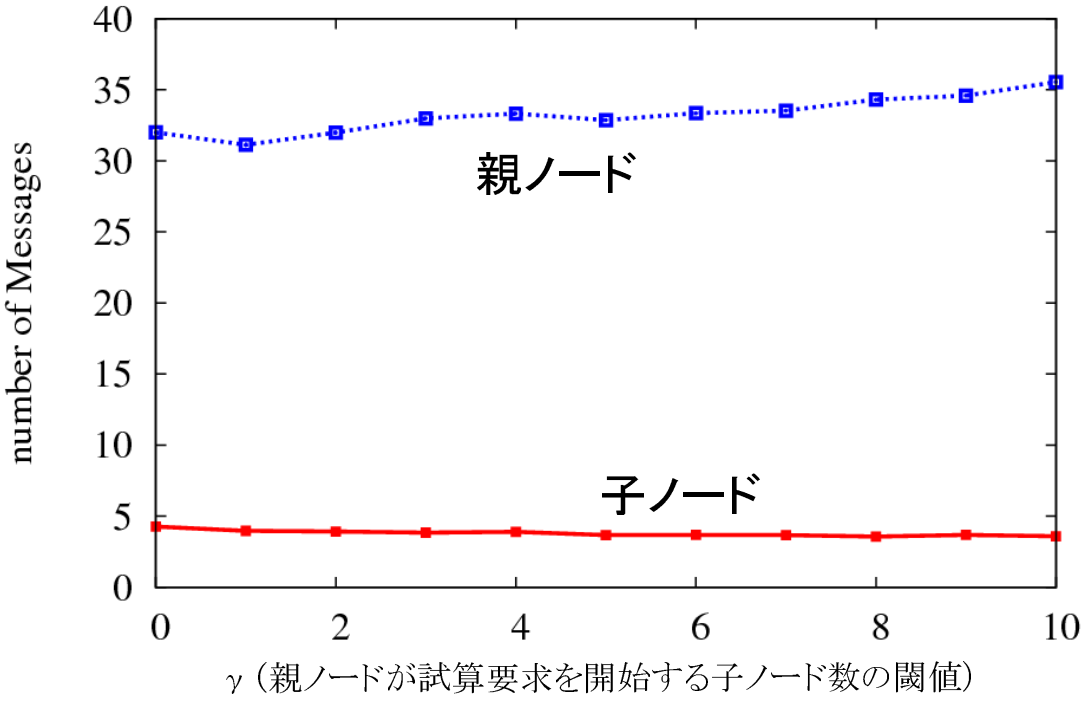

図 4.18: ノード種別ごとの 1 ノードあたりの処理メッセージ数

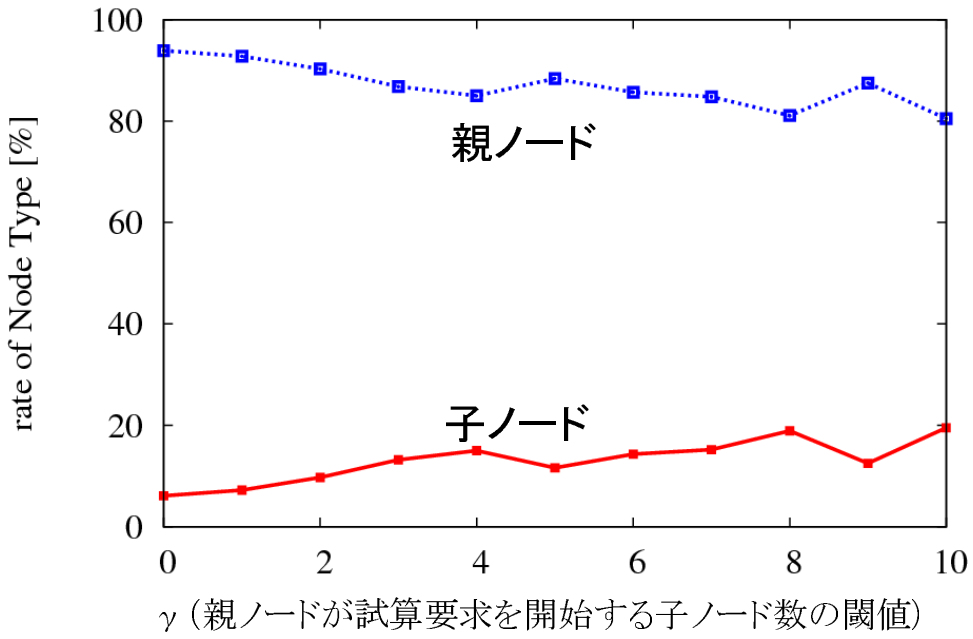

図 4.19: 高性能ノードの親ノードと子ノードの割合の変化

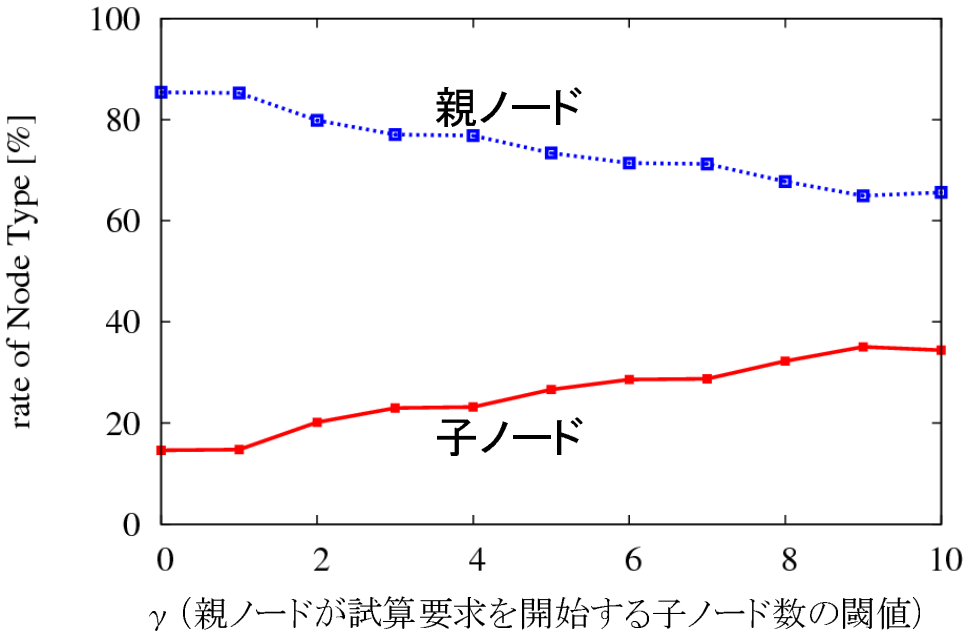

図 4.20: 中性能ノードの親ノードと子ノードの割合の変化

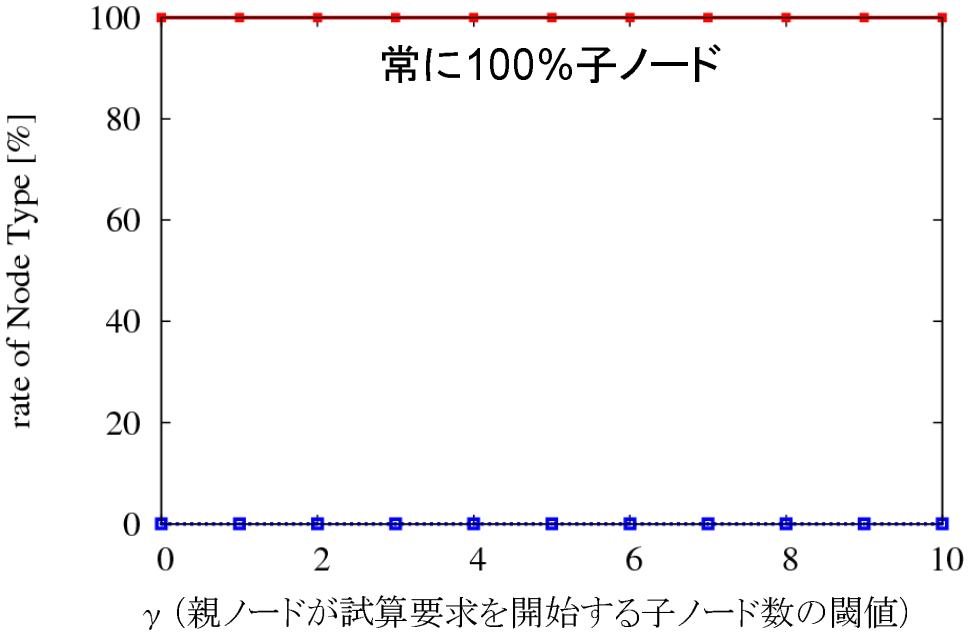

図 4.21: 低性能ノードの親ノードと子ノードの割合の変化

4.3.3 耐障害性評価

提案手法においては，ネットワークに参加する親ノード間で公開鍵の中継を行うことで分散認証を実現しているため，すべての親ノードが正しく動作していることが重要となる．しかし，現実の環境においては，故障などの原因でプロトコル通りに正しく動作しないノードや，悪意を持ってクエリに反応しないノードが紛れ込む危険性がある．そのため，提案手法においては，マルチパスを用いてより確実に目的の公開鍵を得られる工夫を施している．ここでは，提案手法の耐障害性を評価するため，公開鍵の検索における成功率とを計測するとともに，マルチパスを用いた場合に各ノードに生じる処理メッセージ量がどのように変化するかについても確認を行う．

検索成功率

提案手法の耐障害性を評価するため，公開鍵を検索する際の成功率について計測を行った．提案手法で構築するオーバーレイネットワークに 1000 台のノードを参加させ，その中に一定の比率で悪意あるノードが存在しているとし，各ノードにランダムに決定されたノードの公開鍵を取得する手続きを実行させ，その成功率を計測した．本論文における悪意あるノードとは，公開鍵を要求するクエリメッセージに，一切反応しないノードであるとする．図 4.22 に，ネットワークに参加する悪意あるノードの比率と検索の成功率について表したグラフを示す．実験を行ったパスの本数は，1 本，3 本，5 本の 3 種類である．悪意あるノードが存在しない場合においては，パスの本数に関わらず確実に公開鍵を取得可能であることが分かる．また，パスの本数を増加させることで，検索の成功率が向上することが確認できる．しかし，提案手法を安全かつ確実に動作させるためには，マルチパスを用いる場合においても悪意のあるノードの参加を最低限に抑える必要がある．そのためには，運用方法を工夫することで，対策を講じることが可能になると考えられる．具体的な運用方法の例については，後述する

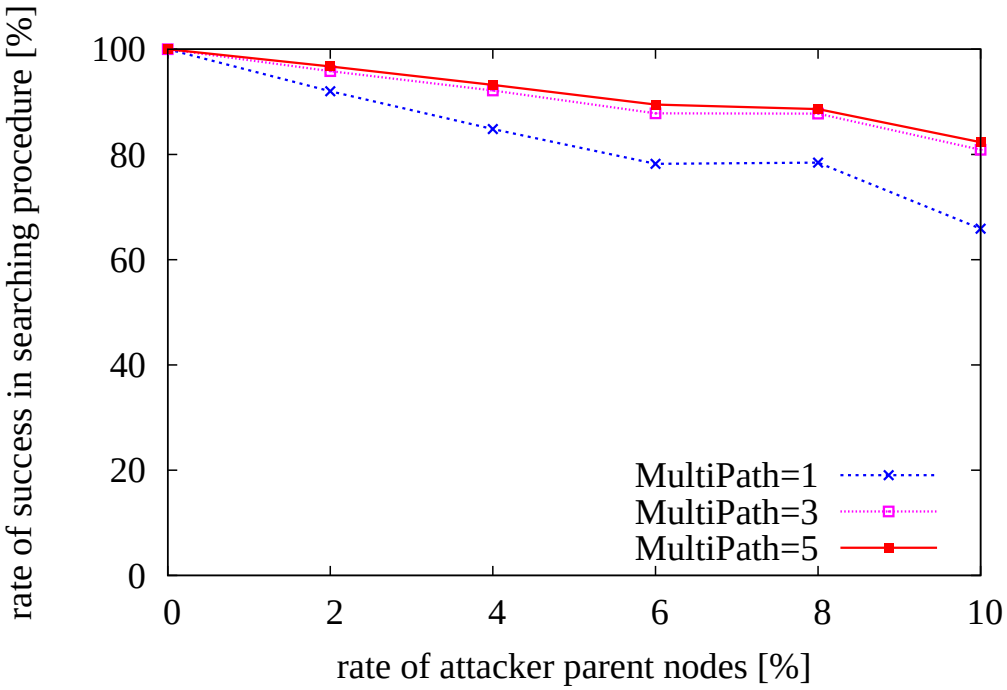

図 4.22: 検索の成功率

マルチパスを用いた場合の通信メッセージ量

提案手法においてマルチパスを用いた場合に，通信データ量がどの程度増加するかを示した結果を，図 4.23 と図 4.24 に示す．これは，認証ネットワークに参加するノード数を 10 台から 1000 台まで増加させていき，すべてのノードが表 4.1 に示した Pattern 1 の確率に従い，図 4.2 に示した状態遷移を 10 回行ったとき，各ノード 1 台あたりが 1step（すべてのノードに状態遷移を 1 回ずつ行わせる）あたりに受信したメッセージ数の平均の個数について計測を行ったものである．図 4.23 は，親ノード 1 台あたりが処理したメッセージ数について示してある．ここで，比較としては S-O-M 手法 [20] を用いた．これは，S-O-M 手法はほぼ確実に必要な公開鍵を得られる手法であるためである．図 4.24 は，提案手法における子ノード 1 台あたりが処理したメッセージ数について示してある．マルチパスの本数を増加させることで，親ノードが処理するメッセージ量は多少増加するものの，既存手法と比較すると，極めて少ない通信データ量であることが分かる．また，子ノードにつ

いては，マルチパスの本数に関わらず，処理メッセージ量を常に極めて少なく抑えることを可能としていることが分かる．よって，提案手法は，マルチパスを用いる場合においても，スケーラブルな分散認証を維持可能であることが分かる．

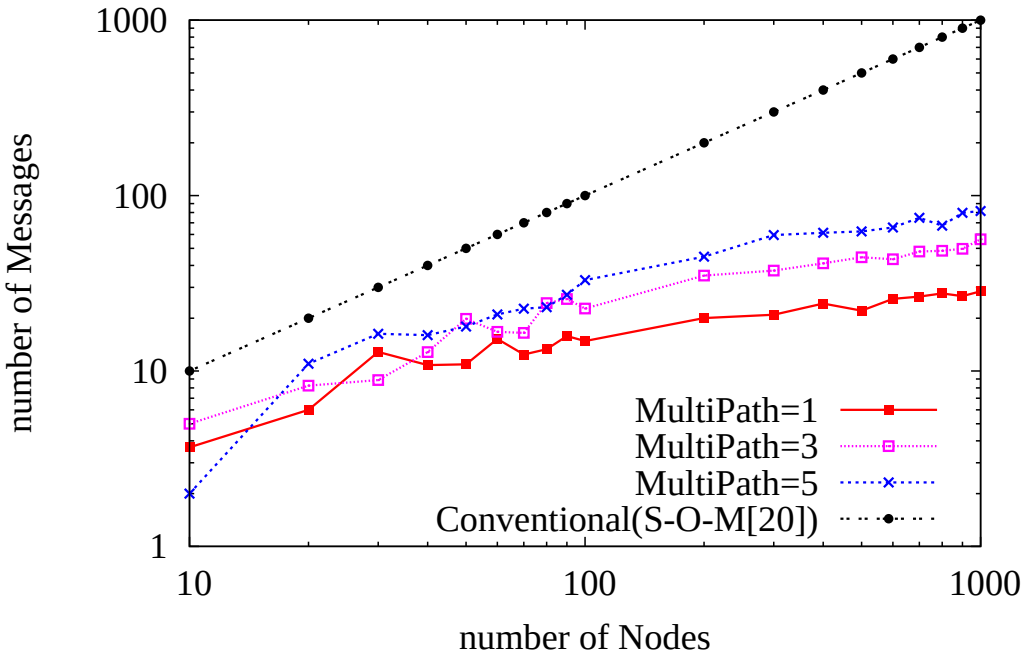

図 4.23: 親ノード 1 台あたりの通信メッセージ数

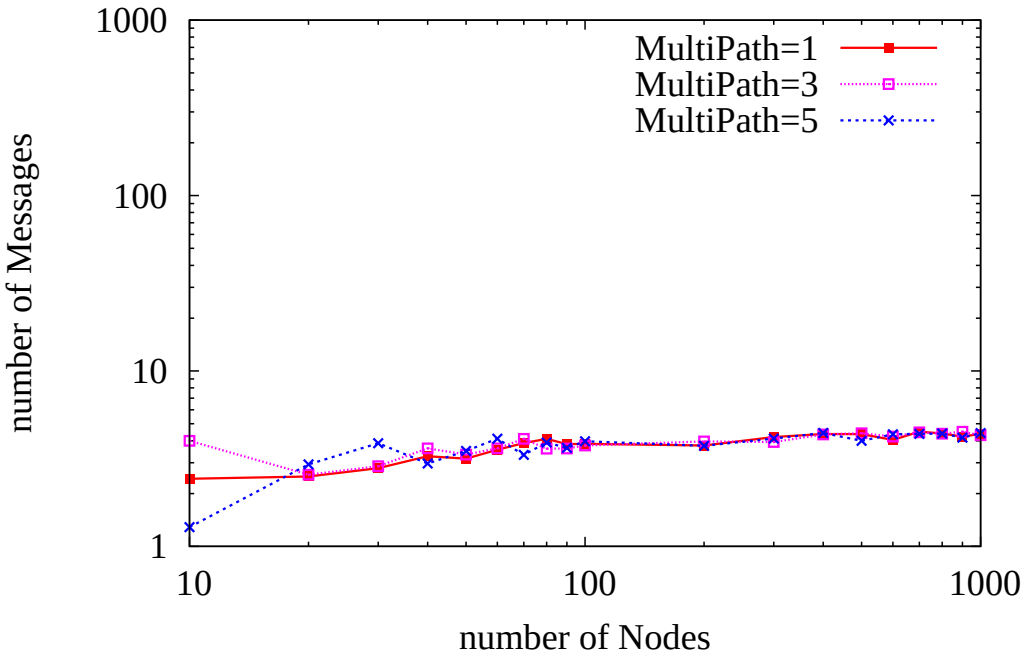

図 4.24: 子ノード 1 台あたりの通信メッセージ数

4.4 考察

本節では，第 4.2 節で述べた評価項目についての考察を行う．続いて，提案手法の適切な運用方法について検討を行い，本研究の成果について述べる．

4.4.1 評価について

（1）性能評価 提案手法は，低性能端末でも参加可能なスケーラブルな分散認証手法を実現することを目的としている．そのため，低性能端末にかかる処理負荷と要求されるメモリ量を最小限に抑えることが重要となる．提案手法により，オーバーレイネットワークに参加する端末を階層化することで，低性能端末に必要となる通信データ量とメモリ量を極めて少量に抑えられることが確認できた．よって提案手法は低性能端末でも参加可能なスケーラブルな分散認証手法であるといえる．すなわち，提案手法はセンサデバイスなどの低性能端末を含む大規模オーバーレイネットワークへの対応が可能な認証基盤であるといえる．

（2）機能確認 提案手法においては，端末の資源状況に応じて動的にオーバーレイネットワークを形成することで，低性能端末や，資源状況に余裕がない端末の負荷を削減することが可能となる．そのため，ネットワークに参加するノードの種別を適切に分類し，オーバーレイネットワークを構築することが重要となる．提案手法により，オーバーレイネットワークに参加する端末の資源状況を測定し，資源状況に余裕がない端末を子ノードとすることで負担を削減可能であることが確認できた．また逆に，資源状況に余裕が見られる端末を親ノードとしてオーバーレイネットワークに参加させることで，一部のノードのみに負荷が集中してしまう事態を最小限に避けることが可能となった．よって提案手法は資源状況に応じて動的にオーバーレイネットワークが構築可能であるといえる．加えて，用いる閾値を適切に設定することで，ネットワークに参加する端末の負荷を最小限に抑えることが可能となり，低性能端末が多数混在した状態においてもスケーラビリティを維持することが可能となるこ

とが分かった．すなわち，提案手法を用いることで，認証のための資源状況に余裕がないノードの判別を行い，その端末の負荷を軽減することが可能である．

（3）耐障害性評価 提案手法において，悪意あるノードが存在しない場合においては，確実に目的の公開鍵を検索可能であることが確認できた．また，悪意あるノードが存在している場合においても，マルチパスを用いて公開鍵の検索を行うことで，検索の成功率が増加することが確認できた．加えて，マルチパスを用いる場合においても，各端末に生じる負荷が極端に増加せず，スケーラビリティに優れていることを確認した．すなわち，提案手法は，ネットワークに悪意あるノードが存在する場合においても，高い確率で公開鍵の検索が可能な認証基盤であるといえる．

以上，性能評価・機能確認・耐障害性評価の３つの評価結果より，提案手法により，安全でスケーラブルな認証基盤が実現可能であることが分かる．よって，提案手法を用いることで，ユビキタス情報環境におけるプライバシー情報を用いたサービスを安全に提供するための基盤を実現することが出来る．

4.4.2 提案手法の運用方法について

オーバーレイネットワーク上で１人の参加者が複数のノードをコントロールすることで，ネットワークが正しく動作しなくなるよう行う攻撃を，Sybil Attack[31] という．一般に Sybil Attack は信頼できる中央認証サーバを用いなければ解決が困難であることが知られている．本研究においては，低コストでスケーラブルな認証基盤の実現を目標としているため，審査などを行う第３者機関を設置することはふさわしくない．しかし，この問題は運用方法を工夫することで解決を図ることが出来ると考えられる．

提案手法によって構築されるオーバーレイネットワークに参加するためには，参加ノードがあらかじめ，先に認証ネットワークに参加していた任意のノードと公開鍵を交換しておくことが条件となっており，そのノードに招待される形でオーバーレイネットワークに参加することとしている．そのため，運用方法として，新しく参加するノードを招待する

権限を持つノードを，社会的に広く知られている信頼性の高いノードに限定することにより，悪意のあるノードがネットワークに参加することを防止することが可能となる．招待権限を持つノードの一例としては，インターネットサービスプロバイダ（ISP）のサーバ端末などが考えられる．

4.5 本研究の成果

本節では，本章で得た評価結果に基づき，本研究の成果についてまとめる．具体的には，アプリケーションサービスなどの開発者にとっての利点と，サービスを提供される一般利用者にとっての利点のそれぞれについて説明する．

4.5.1 サービス開発者に対する成果

計算機端末間で，プライバシーに関わる情報を送受信する場合，公開鍵暗号を用いて安全に通信を行う必要がある．これは現状では，PKI[3] を用いることで実現されている．しかし，PKI においてはサービス提供者側が認証局からサーバ証明書を取得する必要があり，そのためには，複雑な手続きと，高額な費用が必要となる．よって，個人開発者などにとって安全な通信が要求されるサービスの開発は敷居が高く，参入が困難なものであった．本研究で構築する認証のためのオーバーレイネットワークを用いることで，サービス開発者は複雑な手続きや高額な費用を必要とせずに，公開鍵暗号を用いた安全な通信を行うサービスを実現することが可能となる．よって，本研究の成果により，図 4.25 に示すように，多くの開発者が安全な通信を要求されるサービスを積極的に開発し，提供することが可能となる．このことは市場の活性化にもつながると考えられる．

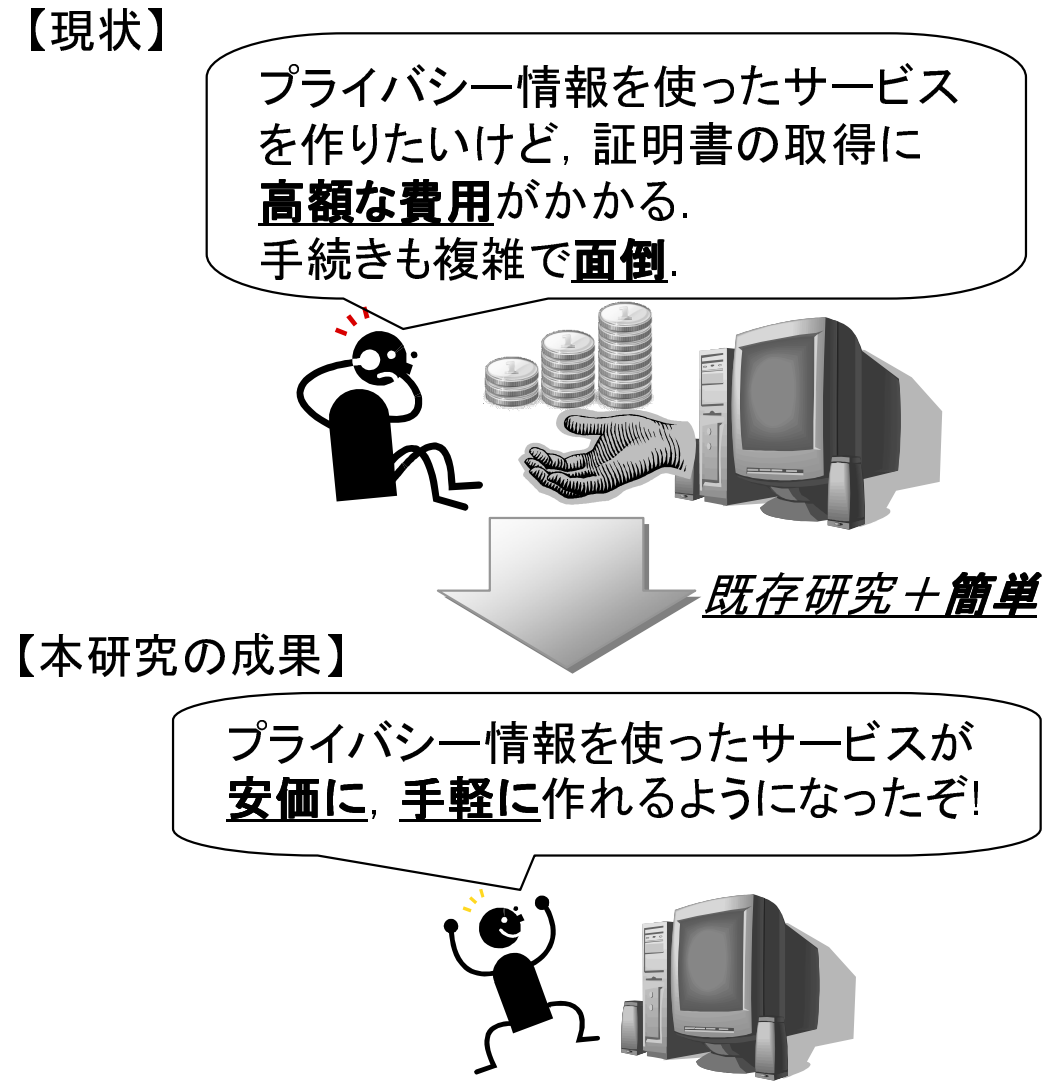

図 4.25: サービス開発者から見た本研究の利点

4.5.2　一般利用者に対する成果

サービスを提供されることを目的として，プライバシー情報をサービス事業者に提供する場合，サービス事業者がサーバ証明書を取得していない場合においては，公開鍵暗号を用いた安全な通信を行うことが出来ない．本研究の成果により，これまで PKI においてサーバ証明書を取得できなかった事業者との通信においても公開鍵暗号を用いた安全な通信を行うことが可能となる．本研究で構築する認証のためのオーバーレイネットワークを用いることで，自身の提供するデータが通信路において漏洩する心配や，通信相手が偽者のなりすましである危険がなくなる．これより，図 4.26 に示すように，プライバシーに関わる情報を安心して事業者に提供することが可能となる．

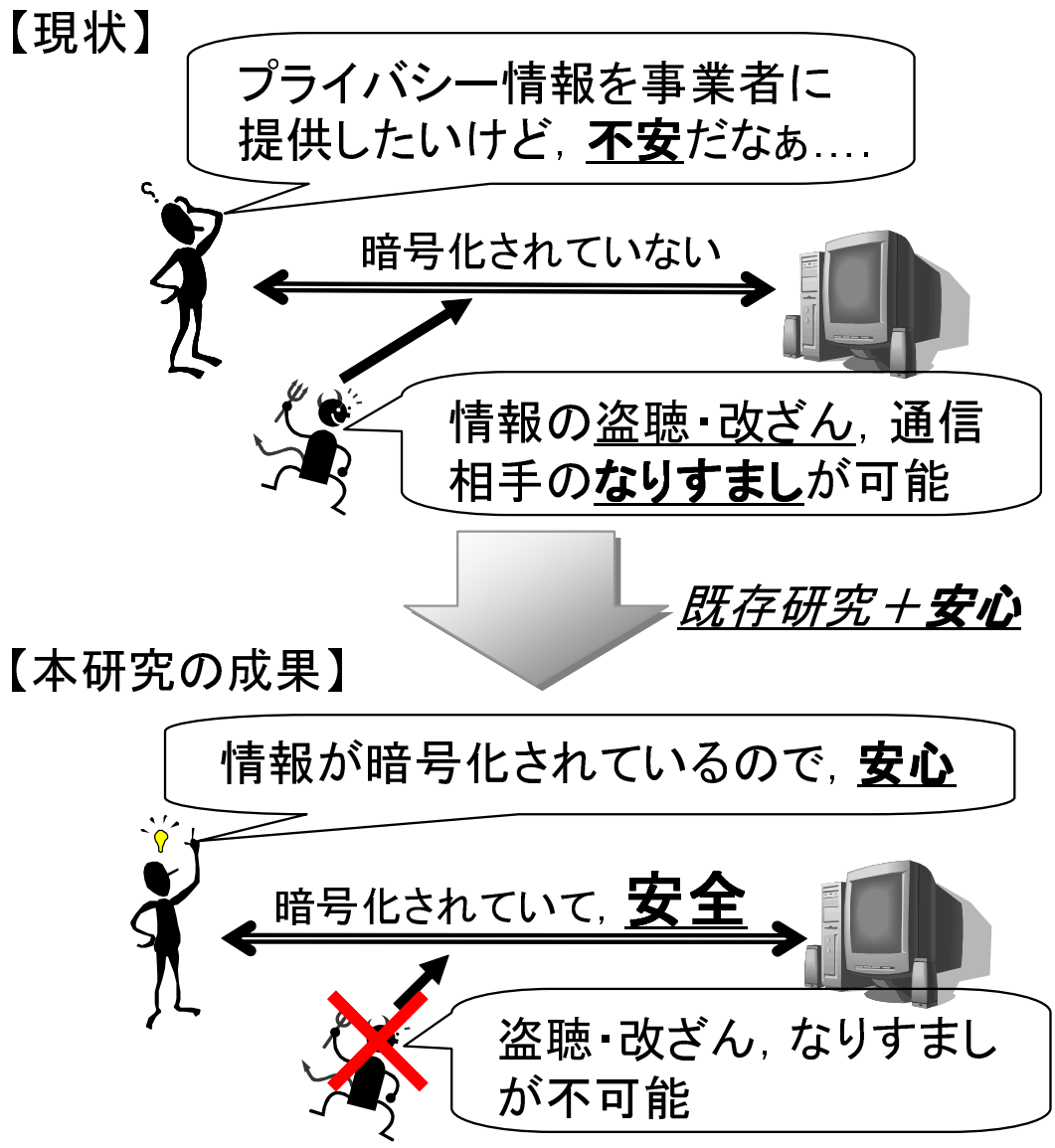

図 4.26: 一般利用者から見た本研究の利点

第5章　結論

5.1　まとめ

本研究の目的は，ユビキタス情報環境において，複雑な手続きなどを必要とすることなく，個人情報などのプライベートな情報を安全に通信可能とすることである．そこで本論文では，認証のためのオーバーレイネットワークを動的に構築する，階層型公開鍵分散認証方式 HiHDAM（Hierarchical Hash-based Distributed Authentication Method）を提案した．本手法は，認証のためのオーバーレイネットワークに参加する端末の資源状況に応じて，動的にオーバーレイネットワークを構成することで，センサデバイスや携帯端末などの計算能力が低い端末であっても要求した公開鍵の確実な配布を受けることが可能となる．また，公開鍵の取得を行う際に，複数の経路を用いて検索を行うことで，オーバーレイネットワーク中に悪意のある端末が紛れ込んでいる状態においても，高い成功率で目的の公開鍵を取得することが可能となる．実験の結果より，本手法を用いて端末間で分散認証を行うことで，従来手法を用いる場合と比較した場合，資源状況に余裕がない端末に生じる負荷が大幅に削減可能であることを示し，よりスケーラブルな認証基盤が実現可能であることを示した．また，公開鍵の取得を行う際に，複数の経路を用いて検索を行うことで，オーバーレイネットワークの耐障害性が向上することを確認した．これらの結果から，提案手法により，安全でスケーラブルな認証基盤が実現可能であることが分かる．よって，提案手法を用いることで，ユビキタス情報環境におけるプライバシー情報を用いたサービスを安全に提供するための基盤を実現することが出来る．

5.2　今後の課題

本研究では，計算機シミュレーションを用いて提案手法の評価を行ったが，今後，ネットワークの構成や遅延などを考慮した，実環境上における実験を通して，その効果を検証する必要がある．また，本手法を誰もが気軽に参加できるという利便性の高さを活かしたまま，安全な認証基盤として運用するためには，適切な運用方法を定めることが重要となる．運用方法を工夫することで，本手法で構築する認証のためのオーバーレイネットワークに悪意あるノードが参加する危険性を最小限に留めることが可能になると考えられる．具体的には，オーバーレイネットワークに新たな端末が参加する際に，新しく参加する端末の招待権限を持つノードを限定するといった方法が考えられる．その際，新しく参加する端末に対し，PKI[3] ほどの複雑な手続きは行わず，最低限必要な本人性の確認を行うようにすることで，利便性を維持したまま，安全な認証基盤として運用することが可能となる．招待権限を持つノードの具体例としては，インターネットサービスプロバイダ（ISP）のサーバ端末などが考えられる．また，本研究では第 2 章 2.2 節で示したオーバーレイネットワークにおけるセキュリティ技術のうち，Web of Trust に焦点をあて，効率的な分散認証の実現を行ったが，Statistical Trust や Hybrid Trust といった技術とあわせて用いることで，本手法を安全な通信基盤として運用しながら，ネットワークから悪意のあるノードを排除するといったことも可能になると考えられる．

加えて，認証のためのオーバーレイネットワークに対して大規模な攻撃が行われた際など，異常発生時における対処策に関しても方針を定める必要がある．提案手法を用いることで，計算機端末が通信相手から認証要求を受けた際に，通信相手の公開鍵を入手し，安全な通信を実現することが可能となるが，認証のためのオーバーレイネットワークに異常が発生した際には，安全な通信を行うことが不可能となってしまう．具体的な対策方針の案として，異常発生時においては，プライバシー情報を送信する際に，利用者に暗号化が行われない旨と，その危険性を提示した上で，送信してもよいかの判断をしてもらうなどといった対策などが考えられる．

また，プライバシー情報を用いたサービスの提供が普及するためには，プライバシー情

報をどのような順番で交換するかなどといった具体的なプロトコルや，プライバシー情報をどこに保存し，どのように扱うかなどといった指針に関しても明確に定めなければならない．クラウドコンピューティングを中心とした仮想化技術の発展により，サービス利用者側は自身が提供した情報がどこに保存され，誰に見られる可能性があるのか，といったことを意識する機会が減少している．しかし，日本ネットワークセキュリティ協会（JNSA）の調査報告書 [32] によると，2008 年の個人情報漏洩事件は 1373 件あり，漏洩人数の合計は 723 万 2763 人とされており，個人情報保護が重要視される流れは，今後ますます強くなってくると考えられる．よって，利便性に優れ，かつ安心して情報を提供することの出来る情報基盤が必要になると考えられる．

謝辞

本論文を終えるにあたり，このような興味深い研究テーマを与えていただき，日頃の研究活動の際に貴重な御指導を賜りました東北大学電気通信研究所教授 白鳥則郎先生に心から感謝いたします.

また，東北大学サイバーサイエンスセンター教授 木下哲男先生には，審査時にとどまらず，大学院セミナー等を通し，日頃から数多くの有益な御指導，御助言を賜りました．心より感謝いたします.

東北大学大学院情報科学研究科教授 橋本和夫先生からも，審査時に加え，日頃行われた研究打ち合わせ等を通し，有益なご指導，ご助言を数多く賜りました．深く感謝いたします.

そして，東北大学電気通信研究所准教授 北形元先生には，本研究をはじめ研究生活全般にわたって終始御指導，御助言を賜りました．心より感謝いたします.

東北文化学園大学講師 武田敦志先生には，常日頃から，大変丁寧なご指導を賜りました．なにも分からなかった私に対して，常に丁寧に相談に応じて頂き，研究の楽しさを教えてくださいました．深く感謝いたします.

東北大学電気通信研究所 白鳥研究室の皆様には，折に振れ貴重な示唆，ご助言をいただきました．また，皆様のおかげで研究生活がより楽しく充実したものとなり，私の精神的な支えとなりました．感謝いたします.

最後に，研究生活の内外で私を支えてくれた多くの友人と，これまで精神的，経済的に支えてくれた家族に深く感謝して，本論文を締めくくりたいと思います.

発表論文

国際会議

1. A. Takeda, S. Nakayama, G. Kitagata, D. Chakraborty, K. Hashimoto, and N. Shiratori, ”Hash-based Distributed Public Key Infrastructure for Ubiquitous Environments” in *Proc. of the 4th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS2010)*, Feb., 2010. (accepted)

研究会・ワークショップ

1. 武田敦志，中山誠也，北形　元，チャクラボルティ デバシシュ，白鳥則郎，　”ユビキタスコンピューティング環境のための分散型公開鍵認証基盤の設計”　情報処理学会シンポジウムシリーズ, マルチメディア，分散，協調とモバイルシンポジウム論文集，　pp. 774-779 2009.07.

2. 中山誠也，武田敦志，北形　元，チャクラボルティ デバシシュ，白鳥則郎，　”ノードの性能を考慮した階層型分散認証手法の設計と評価 ”　平成 21 年度電気関係学会東北支部連合大会講演論文集，2B19，p.69，2009.08.

3. 中山誠也，武田敦志，北形　元，チャクラボルティ デバシシュ，白鳥則郎，　”P2P ネットワークおけるノードを階層化した公開鍵分散管理方式”　FIT2009 講演論文集．2009.09.

4. 中山誠也，武田敦志，北形　元，チャクラボルティ デバシシュ，白鳥則郎，　”公開鍵分散管理を目的としたオーバーレイネットワーク適応型構成法の設計”　電

子情報通信学会 IN 研究会，Vol.109, No.189, IN2009-56, pp.99-104, 2009.09.

5. 中山誠也，武田敦志，北形　元，チャクラボルティ デバシシュ，白鳥則郎，　”ユビキタス情報環境のための階層型公開鍵分散管理方式”　情報処理学会シンポジウムシリーズ, マルチメディア通信と分散処理ワークショップ論文集，　Vol. 2009, No. 9, pp. 195-200, 2009.10.（優秀プレゼンテーション賞）

6. 中山誠也，武田敦志，北形　元，チャクラボルティ デバシシュ，白鳥則郎，　”安全なユビキタス環境のための認証ネットワーク構築法”　日本学術振興会インターネット技術第 163 委員会情報流通基盤分科会 (ITRC/INI)「情報流通基盤分科会ワークショップ」「先端的ネットワーク＆コンピューティングテクノロジワークショップ」合同ワークショップ，2009.10.

7. 半井明大，中山誠也，武田敦志，北形　元，橋本和夫，白鳥則郎，　”プライバシーを考慮した分散認証法の提案”　電子情報通信学会総合大会，2010.3.（発表予定）

参考文献

[1] 田原康生, “1. はじめに : ユビキタスネット社会の実現に向けて (小特集: ユビキタスネットワーク技術開発プロジェクト),” 電子情報通信学会誌, vol. 91, no. 7, pp. 563–568, Jul., 2008.

[2] H. Takahashi, S. Izumi, T. Suganuma, T. Kinoshita, and N. Shiratori, “ An agent-based healthcare support system in ubiquitous computing environments, ”Lecture Notes In Computer Science, vol. 5597, pp. 237?240, 2009.

[3] R.Housley, W.Polk, W.Ford, and D.Solo. Rfc 3280: Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile, 2002.

[4] Xie Dongqing, Neng Jian. PKI principle and technology. Beijing: Tsinghua University Press; 2004.

[5] VeriSign, Inc.， http://www.verisign.com/

[6] 日本ベリサイン株式会社，https://www.verisign.co.jp/

[7] OpenID，http://www.openid.ne.jp/

[8] 岡下 綾，本人確認基盤と公開 ID の提案　～ Web 横断的な匿名本人確認と OpenID の有用性について ～，信学技報, vol. 108, no. 460, IA2008-76, pp. 55-60，2009

[9] Napster，www.napster.com/

[10] Gnutella，http://gnutella.wego.com.

[11] BitTorrent，http://bittorrent.com.

[12] 金子勇，”Winny の技術”，アスキー，2005

[13] Stoica I, Morris R, Liben-Nowell D, Karger D.R., Kaashoek M.F., Dabek F and Balakrishnan H., ”Chord : A Scalable Peer-toPeer Lookup Protocol for Internet Applications”, IEEE/ACM Trans. Networking, Vol.11, No.1, pp.17-32, 2003.

[14] Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Schenker, S., ”A scalable content-addressable network”, Proceedings of the ACM SIGCOMM 2001, pp.161-172, 2001.

[15] J. Aspnes and G. Shah, “ Skip graphs ”, ACM Transactions on Algorithms, vol. 3, no. 4, 2007.

[16] Rowstron, A. I.T. and Druschel, P., ”Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems”, Proceedings of the IFIP/ACM In-ternational Conference on Distributed Systems Platforms Heidelberg (Middleware 2001), Springer-Verlag LNCS 2218, pp.329-350, 2001.

[17] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, “ Tapestry: A resilient global-scale overlay for service deployment, ” IEEE Journal on Selected Areas in Communications, vol. 22, no. 1, 2004.

[18] Zhongwen Li, Xiaochen Xu, Liang Shi, Jian Liu and Chen Liang, “ Authentication in Peer-to-Peer Network: Survey and Research Directions ”, Third International Conference on Network and System Security. pp115-122, 2009.

[19] Simson Garfinkel., ”PGP : Pretty Good Privacy”, Oreilly and Associates Inc., 1994.

[20] Srdjan Capkun, Levente Buttyan, and Jean-Pierre Hubaux. Self-organized public-key management for mobile ad hoc networks. IEEE Transactions on Mobile Computing, 2, No.1:52-64, 2003.

[21] Kitada, Y., Watanabe, A., Sasase, I. and Takemori, K.: On demand distributed public key management for wireless ad hoc networks, Communications, Computers and signal Processing, 2005. PACRIM. 2005 IEEE Pacific Rim Conference on, pp.454-457 2005.

[22] Goold, J. and Clement, D.M. : Improving Routing Security Using a Decentralized Public Key Distribution Algorithm, Internet Monitoring and Protection, 2007, ICIMP 2007, Second International Conference, 2007.

[23] R Chen, W Guo, L Tang, J Hu, Z Chen, " Scalable byzantine fault tolerant public key authentication for peer-to-peer networks " Lecture Notes in Computer Science, 2008

[24] Hongjin Liua,b, Ping Luoa, Daoshum Wang. " A Scalable authentication model based on public keys " Journal of Network and Computer Applications, pp375-386, 2008.

[25] Zhiwei Gao1,2, Jinsheng Faul, Yufeng Jial, Li Zhang. " A Scalable PKI Based on P2P Network, International Journal of Security and its Applications, Vol. 1, No. 2, October, pp47-58, 2007.

[26] Atushi Takeda, Debasish Chakraborty, Gen Kitagata, Kazuo Hashimoto and Norio Shiratori. " Proposal and Performance Evaluation of Hash-based Authentication for P2P Network ". IPSJ Journal, vol.50, no.2, pp.737?749, 2009.

[27] Zhang Y, Li XX, Huai JP , et al ." Access control in peer2to2peer collaborative systems. ", Proceedings of the 25th International Conference on Distributed Computing Systems (IEEE ICDCS '05) Workshop on Mobility of Peer2to2Peer Systems. Ohio : IEEE Computer Society Press , pp835～840, 2005.

[28] Lee H , Kim K." An adaptive authentication protocol based on reputation for peer2to2peer system. ", Proceedings of the Symposium on Cryptography Information Security. pp661～666, 2003.

[29] Karl Aberer, Anwitaman Datta, and Manfred Hauswirth, Efficient,“ Self-Contained Handling of Identity in Peer-to-Peer Systems ”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 16, NO. 7, 858-869, JULY, 2004.

[30] CHEN Fu-sheng .”Research on Decentralized Public Key Infrastructure Based on Peer-to-Peer ”, Master’s thesis, Huazhong University of Science and Technology, 2006

[31] J. Douceur,“ The Sybil Attack ”, IPTPS ’02, pp.251-260, 2002.

[32] NPO 日本ネットワークセキュリティ協会，2008 年 情報セキュリティインシデントに関する調査報告書，2009. http://www.jnsa.org/result/2008/surv/incident/index.html